ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ANÁLISIS, IMPLEMENTACIÓN Y EVALUACIÓN DE UN PROTOTIPO ROUTER DUAL IPV4/IPV6 CON SOPORTE DE QoS E IPsec SOBRE LINUX, USANDO AHP PARA LA SELECCIÓN DEL HARDWARE E IEEE 830 PARA LA SELECCIÓN DEL SOFTWARE.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN

JIMÉNEZ PUMISACHO GLADYS MAGALY
gladysmag03@hotmail.com

PAZMIÑO SANTIN CARLOS ALBERTO
pazmsca@hotmail.com

DIRECTOR: MSc. XAVIER CALDERÓN
xavieralex_calderon@hotmail.com

Quito, noviembre 2009

DECLARACIÓN

Nosotros, Carlos Alberto Pazmiño Santin, Gladys Magaly Jiménez Pumisacho, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____ _____

Carlos Alberto Pazmiño Santin Gladys Magaly Jiménez Pumisacho

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Carlos Alberto Pazmiño Santin y Gladys Magaly Jiménez Pumisacho, bajo mi supervisión.

MSc. XAVIER CALDERÓN

DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Primero quiero agradecer a Dios por ser quien ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día a día y seguir adelante rompiendo todas las barreras que se me presentan.

Un agradecimiento muy especial al MSc. Xavier Calderon por su inmensa colaboración durante la realización del presente Proyecto de Titulación.

A mis amigos más cercanos, a esos amigos que siempre me han acompañado en momentos buenos y malos y con los cuales he contado desde que los conocí.

A todos los profesores y a todas las personas que influyeron de una u otra forma para alcanzar esta meta.

Carlos Alberto

AGRADECIMIENTO

Un especial y sincero agradecimiento, al MSc. Xavier Calderón, por su colaboración profesional y desinteresada que me brindó para la realización de este Proyecto de Titulación.

Agradezco a mis padres por el constante apoyo moral durante todo el trayecto de mi carrera y a mi familia por brindarme siempre su cariño y apoyo incondicional.

Agradezco a mis compañeros y amigos por su compañía, consejos y su amistad sincera.

Un agradecimiento muy especial a todos aquellos profesores que me guiaron para la culminación de esta meta y a todas aquellas personas de una u otra manera fueron un aliciente para la culminación de este proyecto.

Para ellos,

Muchas gracias por todo.

Gladys Magaly

DEDICATORIA

Dedico este proyecto de titulación y toda mi carrera universitaria a mi madre y a mi abuelita ya que gracias a ellas soy quien soy el día de hoy, ellas con ese cariño y calor humano, fueron las que velaron por mi salud, alimentación, estudio, educación, entre otros, son a ellas a quien les debo todo, muchas horas de consejos, regaños, tristezas y alegrías de las cuales estoy muy seguro que las han hecho con todo el amor del mundo para formarme como un ser integral y de las cuales me siento extremadamente orgulloso.

A mis hermanos los cuales han estado a mi lado, han compartido todos esos secretos y aventuras que solo se pueden vivir entre hermanos y que han estado siempre alerta ante cualquier problema.

Carlos Alberto

DEDICATORIA

Dedico esta tesis a Dios, por estar conmigo en los momentos más difíciles y mostrarme el camino correcto, así como el haberme brindado la perseverancia necesaria para finalizar una meta más en mi vida.

A mis padres Augusto y Yolanda por confiar en mí, por brindarme siempre su cariño y apoyo, a mis queridos hermanos por sus palabras de aliento y fortaleza.

A mis amigos por su compañía y consejos.

Gladys Magaly

CONTENIDO

TOMO I

TOMO II

ÍNDICE DE TABLAS

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

ÍNDICE DE FIGURAS

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

RESUMEN

El presente proyecto de titulación provee una solución de bajo costo para empresas que desean adquirir equipos de enrutamiento que cumplan las características técnicas necesarias, pero con un costo más bajo a los equipos disponibles en la actualidad.

El primer capítulo se realizó un estudio de los algoritmos de enrutamiento y de los principales protocolos utilizados actualmente, además se realizó un análisis del protocolo IP en sus dos versiones IPv4 e IPv6.

Con el fin de priorizar el tráfico más relevante se procede a realizar un estudio para la implementación de calidad de servicio QoS, y finalmente para solucionar el problema de seguridad en capa red se llevó a cabo el estudio del protocolo IPsec.

En el segundo capítulo se realizó un estudio de las distribuciones de Linux más importantes, con el objetivo de seleccionar la más adecuada como base para la implementación de software de enrutamiento que permita establecer un ambiente propicio, estable y confiable para levantar servicios de enrutamiento.

La selección tanto de la plataforma Linux y del software de enrutamiento se realizó en base a la Norma de Selección IEEE 830.

La seguridad y la calidad de servicio también fueron analizadas en este capítulo, por lo que también se realiza un análisis de las opciones más importantes para la implementación de protocolos de seguridad IP y de calidad de servicio QoS.

El primer tema tratado en el tercer capítulo es la selección de hardware mediante el Proceso de Análisis Jerárquico AHP. Luego se procede a la implementación y

configuración del prototipo router dual IPv4 e IPv6; y se realizan las respectivas pruebas de funcionamiento, para determinar el análisis de desempeño del mismo.

En el cuarto capítulo se genera una estimación del costo del prototipo implementado, y se realiza una comparación técnico – económica con ruteadores de marcas reconocidas en el mercado que presenten características similares a las del prototipo.

Con los conocimientos adquiridos y detallados en los anteriores capítulos, en el quinto capítulo se presentan las conclusiones y recomendaciones obtenidas del proyecto.

Por último se presentan los anexos que dan soporte al proyecto de titulación.

PRESENTACIÓN

El presente proyecto tiene como finalidad proporcionar una solución de bajo costo a empresas que desean adquirir equipos de enrutamiento que cumplan las características técnicas, pero con un costo más bajo a los equipos disponibles en la actualidad.

El sistema operativo GNU/Linux presenta interesantes posibilidades en el ámbito del enrutamiento, pudiendo realizar configuraciones muy simples o muy complejas, de acuerdo a las necesidades del usuario, por lo que, no siempre que se necesite un enrutador es necesario adquirir el hardware específico si no que se puede conseguir la misma funcionalidad pero por medio de configuración de software.

Existe la posibilidad de montar un enrutador sobre un sistema operativo libre y gratuito como es GNU/Linux. De hecho, en manos expertas, se pueden configurar redes realmente complejas, por lo que esto, también genera una alternativa a la adquisición de equipo de uso específico como son los enrutadores, teniendo en cuenta los altos costos de estos equipos, y el problema que genera la adquisición de actualizaciones para la implementación de protocolos nuevos, por lo que este proyecto pretende generar una alternativa de bajo costo y con mayores posibilidades de actualización de nuevos protocolos.

También permitirá la implementación de protocolos de seguridad a través del protocolo IPsec, y priorización de tráfico por medio de reglas de calidad de servicio proyectado básicamente para empresas que están en proceso de crecimiento y manejen escenarios específicos en los que el enrutador GNU/Linux pueda proponer una solución técnica y financieramente viable, y con el respaldo

científico y técnico dado por las normas IEEE 830 y el proceso de análisis jerárquico (AHP).

CAPÍTULO 1 ESTUDIO DE LOS PROTOCOLOS A SER IMPLEMENTADOS

1.1 INTRODUCCIÓN

El envío de información a través de una red debe ser de manera confiable y sin pérdidas de datos, existen varios elementos (software-hardware) que posibilitan este intercambio de información.

Una red de datos comprende una o más computadoras conectadas entre sí las cuales comparten recursos como impresoras, áreas de almacenamiento, e información como archivos, además constan de equipos de comunicaciones cuyo funcionamiento es transparente al usuario, todos estos equipos deben ser capaces de identificarse entre sí y brindar conectividad a cortas y largas distancias, entre las ventajas que proporciona una red de datos es importante mencionar la posibilidad de crecimiento y la integración de varios puntos en un mismo enlace además de no estar limitada a un espacio geográfico.

Entre los componentes de una red se encuentra el servidor el cual ejecuta el sistema operativo de red, y ofrece a las estaciones de trabajo los servicios de red. Las estaciones de trabajo pueden ser computadoras personales con el DOS, Macintosh, Unix, OS/2 o estaciones de trabajos sin discos; otros de sus elementos son las tarjetas de red, los medios de propagación (guiados o no guiados) y los equipos de conectividad como por ejemplo los hubs, bridges, switches, modems y ruteadores.

Un ruteador o encaminador es un dispositivo de hardware que opera a nivel de capa red su función es proporcionar la interconexión entre computadoras asegurando el enrutamiento de paquetes entre las redes, existe la posibilidad de no usar estos equipos e implementar un software que maneje protocolos para distribuir el tráfico eficientemente, esto consiste en adaptar a un PC con sistema operativo Linux la funcionalidad de enrutador, para este propósito

basta añadir al menos dos interfaces de red y utilizar paquetes específicos que soporten varios protocolos de red, ésta es una enorme ventaja ya que el uso de equipos de casas fabricantes presentan un buen rendimiento pero a un costo elevado.

En el presente capítulo se realizó un estudio de los algoritmos de enrutamiento y de los principales protocolos utilizados en la actualidad, además se realizó un análisis del protocolo IP en sus dos versiones IPv4 e IPv6 y se procedió a desarrollar una comparación en cuanto a las prestaciones que brindan cada uno.

Con el fin de priorizar el tráfico más relevante se procede a realizar un estudio de la implementación de calidad de servicio QoS. Finalmente para solucionar el problema de seguridad en capa red se llevó a cabo el estudio del protocolo IPsec.

1.2 ENRUTAMIENTO

El enrutamiento permite determinar la mejor ruta para enviar la información. En redes de comunicaciones existen múltiples caminos para llegar a un mismo destino esta función nos permite escoger el mejor de ellos basado en tablas de enrutamiento de cada ruteador.

Para este propósito se implementan protocolos de enrutamiento, es necesario recordar que su funcionamiento es transparente para el usuario.

1.2.1 ENRUTAMIENTO ESTÁTICO

El enrutamiento estático es configurado por el administrador de red y básicamente se lo utiliza cuando la red presenta pocos ruteadores en su infraestructura, su desventaja es que no permite una actualización automática de la ruta.

Entre sus principales ventajas es el no consumo de ancho de banda de la red y la seguridad que proveen ya que únicamente el administrador de red configura las rutas.

1.2.2 ENRUTAMIENTO DINÁMICO

El enrutamiento dinámico es utilizado en redes grandes, permite actualizar la tabla de rutas debido a cambios en la red o al tráfico en la misma. Si se presentan varias alternativas hacia un mismo destino o si existe fallas en la ruta principal los protocolos de enrutamiento dinámico se encargan de tener una ruta de respaldo que garantice la llegada correcta de los paquetes a su destino final.

Es importante recalcar que el administrador optimiza tiempo al usar enrutamiento dinámico, por ejemplo al interconectar varias redes con un sin número de ruteadores si se presenta algún cambio en la topología de la red, el administrador tardaría demasiado tiempo en configurar cada ruteador.

Como ya mencionamos sus ventajas son la actualización automática, ahorro económico ya que no es necesario el trabajo del administrador, sin embargo los requerimientos de ancho de banda se incrementan a medida que aumenta las tablas de enrutamiento en cada router.

1.2.3 DEFINICIONES

Son necesarios algunos conceptos para comprender mejor el presente capítulo.

- Tiempo de Convergencia

  Conviene que el tiempo de convergencia sea el menor posible ya que representa el tiempo que tardan los ruteadores en recalcular las rutas cuando ha variado la topología de la red.

Decimos que existe convergencia cuando los ruteadores conocen y tienen una visión completa de la topología de red, es decir, de las rutas para alcanzar el destino, este tiempo de convergencia varía dependiendo de cada protocolo de enrutamiento.

- Sistema Autónomo

  Un sistema autónomo es un conjunto de enrutadores que manejan la misma política de enrutamiento, el Internet es una colección de sistemas autónomos.

  Un Sistema Autónomo es un conjunto de redes y enrutadores que se encuentran administrados por una sola entidad y cuentan con una política común de definición de trayectorias para Internet. [7]

- Métrica

  La métrica es la información usada para escoger el mejor camino para el enrutamiento, es un valor dado por el ancho de banda, costo de la comunicación, retardo, número de saltos, carga, MTU Maximum Transfer Unit - Unidad Máxima de Transferencia, costo de ruta, y confiabilidad.

  Cada protocolo de enrutamiento calcula la métrica de la ruta hacia la red destino.

- Enrutador

  Enrutador, ruteador o encaminador es un dispositivo de hardware para interconexión de red de computadoras que opera a nivel de la capa tres (nivel de red) del modelo OSI. Este dispositivo permite asegurar el enrutamiento de paquetes entre redes o determinar la ruta que debe tomar el paquete de datos.

- Protocolos de Enrutamiento

  Los protocolos de enrutamiento permiten a los enrutadores intercambiar información de encaminamiento de esta forma se logra conocer la

topología de la red y tomar decisiones óptimas a la hora de encaminar un paquete.

- Demonio

  Un Demonio (daemon en inglés) es un script, un proceso que normalmente está cargado en memoria esperando una señal para ser ejecutado.

- Calidad de servicio (QoS)

  Son las tecnologías que garantizan la transmisión de cierta cantidad de datos en un tiempo dado. Calidad de servicio es la capacidad de dar un buen servicio.

- IPsec

  IPsec abreviatura de Internet Protocol security es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos. IPsec también incluye protocolos para el establecimiento de claves de cifrado.

1.2.3.1 **Tipos de Protocolos de Enrutamiento**

Como hemos mencionado los protocolos de enrutamiento se usan entre routers para generar y actualizar sus tablas de enrutamiento dinámicamente, los datos de usuario no viajan en estos protocolos, es necesario configurar las interfaces y recordar que si se configura un protocolo en determinado ruteador todos deben presentar la misma configuración.

Los protocolos de enrutamiento se dividen en Protocolos de enrutamiento exterior (EGP) e interior (IGP). Entre los protocolos de enrutamiento interior constan los que implementan el Algoritmo Vector-Distancia, los protocolos basados en el Algoritmo Estado de enlace y finalmente los híbridos, estos son utilizados para enrutar dentro de un mismo sistema autónomo. Los EGP sirven

para enrutar entre sistemas autónomos, entre ellos constan el EGP sustituido actualmente por el BGP.

1.2.3.1.1 Algoritmo Vector-Distancia

Los ruteadores que operan con este algoritmo se pasan periódicamente copias de las tablas de enrutamiento hacia los ruteadores vecinos y acumulan vectores-distancia. Estas actualizaciones periódicas entre ruteadores informan los cambios de topología. Este algoritmo también se conoce como algoritmo Bellman-Ford.

El enrutamiento por vector-distancia mantiene una entrada que consta de dos partes: la línea preferida por la que se dirige el paquete para llegar al destino (dirección) y el tiempo que emplea en alcanzarlo (métrica) la cual puede ser:

- Número de saltos.
- Ancho de banda.
- Retardo.
- Confiabilidad.
- Carga.
- Costo (métrica compuesta).

Este algoritmo presenta ciertas desventajas ya que al enviar en cada actualización toda la tabla de ruteo se produce una convergencia lenta de la red, además al definir un número de saltos no diferencia entre redes con mayores prestaciones en velocidad de transmisión y confiabilidad.

Los problemas que se presentan en estos protocolos son los lazos de enrutamiento producido por el tiempo de convergencia excesivo de los routers cuando actualizan su información de enrutamiento, es por ello que se genera información errónea para alcanzar un determinado destino. Esta convergencia

lenta de la red genera que los ruteadores envíen sus tablas de enrutamiento incorrectas a los ruteadores vecinos. Y el conteo al infinito, producido por una convergencia alta ante buenas noticias pero lenta frente a respuestas incorrectas.

Las soluciones posibles son:

- Definición de un máximo número de saltos para prevenir los lazos infinitos, con esta corrección se obliga que cualquier red que este más allá de un determinado número de saltos se considere inalcanzable.
- Split Horizon - Horizonte dividido, aquí se restringe el envío de paquetes a un enrutador desde la dirección de dónde provino la información original de enrutamiento a menos que tenga una mejor métrica.
- Poison Reverse - Reversa a Infinito, cuando un enlace a determinado destino falla los ruteadores configuran su distancia a infinito para evitar enrutamiento erróneo.
- Disparo de Actualizaciones, al variar la topología de la red se envía mensajes para que cada enrutador actualice su tabla de enrutamiento.
- Temporizadores de espera, se aguarda determinados segundos hasta que la red vuelva a estar operante, con ello se recalculan las rutas y se tiene información correcta en la tabla de enrutamiento de los ruteadores.

1.2.3.1.2 Algoritmo Estado de Enlace

En este algoritmo los ruteadores calculan en paralelo la distancia más corta a los destinos, para ello se divide la red en áreas, cada ruteador se encarga de descubrir el estado de enlace con los ruteadores cercanos y ver el costo de la ruta, esta información se distribuye a los ruteadores pertenecientes a cada área; se debe mencionar que cuando un paquete atraviesa múltiples áreas siempre utilizará la ruta más corta para alcanzar su destino.

Cada ruteador envía una copia de un paquete para que sus vecinos tengan una representación general de la topología de la red y para el cálculo de la ruta más corta a su destino.

En este algoritmo se envían Link-State Packets – Paquetes de Estado de Enlace (LSA y LSU), al encender los ruteadores envían LSA Link State Advertisements - eventos de estado de enlace para tener una visión general de la topología de la red, y usan LSU Link State Updates – actualizaciones de estado de enlace para informar al resto de ruteadores que existe un cambio en la topología de la red y deben proceder a actualizar sus tablas de ruteo.

Los problemas que se presentan son las actualizaciones no sincronizadas las que ocasionan que los routers tomen decisiones de caminos inconsistentes y la necesidad de recursos en los ruteadores. Las posibles soluciones consiste en particionar la red en áreas, usar marcas de tiempo en las actualizaciones, regular la frecuencia de las actualizaciones o usar actualizaciones tipo multicast. Otra solución viable consiste en intercambiar sumarios (resúmenes) de rutas en los extremos.

1.2.3.2 **Comparación de Algoritmos**

Algoritmo Vector-Distancia	**Algoritmo Estado de Enlace**
Visualizan la topología de la red desde la perspectiva de los vecinos.	Obtienen una visión común de toda la topología de la red.
Agregan vectores-distancia de ruteador a ruteador.	Calcula la distancia más corta hacia los otros ruteadores (Algoritmo SPF).
Actualizaciones frecuentes, periódicas.	Actualizaciones disparadas por eventos.

Convergencia lenta.	Convergencia rápida.
Envía copias de las tablas de enrutamiento entre los ruteadores vecinos.	Envía actualizaciones de estado de enlace (LSU) hacia otros ruteadores.
Necesitan menos recursos de memoria y capacidad de procesamiento en los ruteadores.	Tienen mayores requerimientos de memoria y capacidad de procesamiento.
	Consumen menos ancho de banda de la red, excepto al inicio que se produce una inundación de LSA en la red, o cuando se produce un cambio en la topología de la red.

Tabla 1- 1 Comparación de Algoritmos de Enrutamiento

1.2.3.3 **Protocolos de Enrutamiento Híbrido**

Combinan las funcionalidades de los protocolos Vector-Distancia y los de Estado de enlace, para determinar la mejor ruta emplean el algoritmo vector-distancia, y al surgir cambios en la topología de la red envían actualizaciones a cada ruteador para que actualice la base de datos de enrutamiento. Emplean menor ancho de banda, memoria y capacidad de procesamiento y proveen una convergencia rápida a cambios de la red.

1.3 PROTOCOLOS DE ENRUTAMIENTO DINÁMICO

Los protocolos de enrutamiento se encargan de descubrir los diferentes cambios de la red y enviar información de ruteo para el encaminamiento

correcto de los paquetes de datos. Son implementados en los ruteadores para actualizar las tablas de enrutamiento debido a cambios en la red y para el aprendizaje correcto de las rutas por las que atravesará el datagrama IP.

1.3.1 RIP - ROUTING INFORMATION PROTOCOL - PROTOCOLO DE ENCAMINAMIENTO DE INFORMACIÓN

Es un protocolo vector-distancia que calcula la ruta más corta para llegar a su destino, es ampliamente difundido e implementado por fabricantes de tecnología de networking.

Fue diseñado por XEROX y definido en el RFC 1058 y en STD56 (documento estándar de Internet).

1.3.1.1 Funcionamiento de RIP

Los paquetes RIP son transmitidos usando datagramas UDP (User Datagram Protocol) a través del puerto 520, es decir utiliza una comunicación no orientada a conexión.

Implementa una métrica basada en el número de saltos, y emplea participantes activos y pasivos para su funcionamiento, los activos cada 30 segundos difunden su vector-distancia compuesto por su dirección IP y la distancia para alcanzar su destino.

Los participantes pasivos actualizan las tablas de enrutamiento debido a que escuchan los mensajes RIP.

Implementan dos tipos de paquetes: Request - Petición y Response - Respuesta.

- Petición: enviados por los routers o host que acaban de conectarse o su información ha caducado.

- Respuesta: Enviados periódicamente, en respuesta a una petición o cuando cambia algún coste[1].[1]

RIP presenta una distancia administrativa[2] de 120, su métrica como mencionamos se basa en el número de saltos definidos en un valor de 15, esto con la finalidad de evitar lazos de encaminamiento, cualquier nodo por el que tenga que realizarse 16 saltos se considera inalcanzable.

RIP procede a borrar rutas si no percibe ningún mensaje de actividad en ellas dentro de 180 segundos, es decir cada ruta tiene un tiempo de vida de 180 segundos antes de ser descartada por RIP.

1.3.1.2 **RIP Timers**

RIP utiliza una gran cantidad de relojes para regular su performance. Estos relojes llamados como timers son:

Routing Update Timer: Es considerado como el tiempo que le toma a cada ruteador para enviar su tabla de enrutamiento a sus ruteadores vecinos, este tiempo se define en 30 segundos.

Route Invalid Timer: Primero se revisa si las rutas no funcionan las pone como no válidas y espera un tiempo para eliminarlas, el tiempo de espera sin haber escuchado a la ruta se estipula en 90 segundos.

Route Flush Timer: Indica el tiempo antes de eliminar una ruta su valor es de 270 segundos.

---

[1] Costo del enlace está relacionado exclusivamente con el ancho de banda (Costo= cte/AB).

[2] Distancia administrativa indica el grado de confiabilidad de un protocolo, a menor valor mejor es el protocolo a ser usado.

1.3.1.3 **Formato de paquete RIP**

Figura 1- 1 Formato de Paquete RIP

- Comando: Especifica si el paquete es una petición o respuesta, consta de 8 bits.

  Petición = 1

  Respuesta = 2

- Versión: Formado por 8 bits e indica a que versión de RIP corresponde el paquete a ser enviado.

  RIPv1 = 1

  RIPv2 = 2

- Identificador de la familia de direcciones: Permite llevar información de enrutamiento de diferentes protocolos, consta de 16 bits.

  IP = 2

- Dirección IP: Contiene la dirección IP de una entrada de la tabla de enrutamiento y está formado por 32 bits.
- Métrica: Contiene el número de saltos que realiza el paquete para llegar a su destino (1-15). Está compuesto por 32 bits.

1.3.1.4 **Controles de estabilidad de RIP**

Los lazos de enrutamiento y la cuenta al infinito son los problemas que presentan los protocolos que se basan en el algoritmo vector-distancia.

Para solucionar estos problemas RIP incorpora los siguientes controles de estabilidad:

- Límite del número de saltos: se define en un máximo de 15 saltos, fuera de ese valor la red es inalcanzable.
- Timers: para controlar las actualizaciones de las tablas de enrutamiento y la eliminación de las rutas no válidas.
- Horizonte dividido.
- Poison Reverse.

1.3.1.5 **Ventajas**

- Sencillo.
- Muy utilizado.
- Mensaje pequeño y de formato uniforme.
- Distribuido por UNIX BCD (router).

1.3.1.6 **Desventajas**

- Diferencias entre implementaciones (superable).
- Convergencia lenta la cual produce inconsistencias en el enrutamiento.
- Lazos de enrutamiento.
- Carga las redes.
- Un mínimo número de saltos 15.
- No se considera retardos, ancho de banda, carga en la redes ya que solo se basan los ruteadores en la métrica.
- Esta versión de RIP no soporta Subnetting – subneteo.
- No soporta máscaras de tamaño variable.

1.3.2 RIP VERSIÓN 2

Esta versión de RIP soporta subredes VLSM Variable-Length Subnet Masking - Máscara de Subred de Longitud Variable, CIDR Classless Inter-Domain

Routing - Encaminamiento Inter-Dominios sin Clases, maneja autenticación con los siguientes mecanismos: no autentificación, autentificación mediante contraseña, autentificación mediante contraseña codificada con MD5.

Envía actualizaciones de tablas de enrutamiento en multicast a la dirección 224.0.0.0, como todo protocolo que usa el algoritmo vector-distancia tiene problemas con los lazos de enrutamiento, además una de sus restricciones son los 15 lazos permitidos para un paquete enviado.

Otra de sus limitaciones es que no permite balanceo de carga únicamente acepta una ruta por cada destino y la generación de mucho tráfico en la red al enviar toda su tabla de enrutamiento en cada actualización.

1.3.2.1 **Formato de paquete RIPv2**

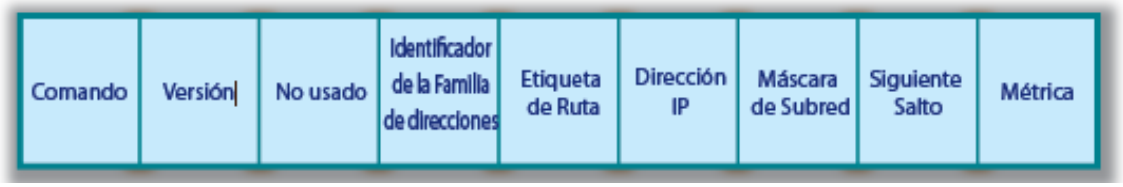

Figura 1- 2 Formato de Paquete RIPv2.

- Comando: Especifica si el paquete es una petición o respuesta, consta de 8bits.

  Petición = 1

  Respuesta = 2

- Versión: Indica a que versión de RIP corresponde el paquete a ser enviado y está formado por 8 bits.

  RIPv2 = 2

- No-usado: Se coloca un valor de 0 en este campo.
- Identificador de la familia de direcciones: Permite definir el tipo de direcciones que se utiliza, consta de 16 bits.

- Etiqueta de ruta: Permite distinguir entre rutas internas y externas. Rutas internas se refiere a rutas aprendidas por este protocolo, rutas externas son consideradas a aquellas aprendidas por otros protocolos y está formado por 16 bits.
- Dirección IP y máscara de subred: Especifican la ruta hacia alguna red destino, cada campo está formado por 32 bits.
- Siguiente salto: Especifica el siguiente salto que debe cruzar el paquete, es una recomendación que indica la dirección IP siguiente a ser alcanzada, formado por 32bits.
- Métrica: Contiene el número de saltos que realiza el paquete para llegar a su destino (1-15) y está constituido por 32 bits.

1.3.3 RIPNG - RIP NUEVA GENERACIÓN

RIPng es la nueva versión de RIP utilizado en redes IPv6, su especificación se encuentra en el RFC 2080, una red global como el Internet está organizada por un conjunto de sistemas autónomos en donde se especifican los protocolos de ruteo a ser utilizados dependiendo de la entidad administradora, RIPng trabaja como un IGP en Sistemas Autónomos de tamaños pequeños, emplea básicamente el algoritmo vector-distancia con sus limitaciones como el número de saltos (1-15) y la imposibilidad de seleccionar las rutas ya que utiliza métricas fijas.

1.3.3.1 **Formato**

Los ruteadores que implementan RIP tienen el prefijo IPv6 del destino, el valor de la métrica es decir el costo que se requiere para alcanzar un determinado destino, la dirección IPv6 del próximo ruteador a ser alcanzado así como la ruta para llegar a él, una bandera que me permite ver los cambios recientes en las rutas y sus respectivos Timers en sus tablas de ruteo.

RIP utiliza el puerto 521 UDP para enviar y recibir datagramas, en la Figura 1-3 se muestran sus campos:

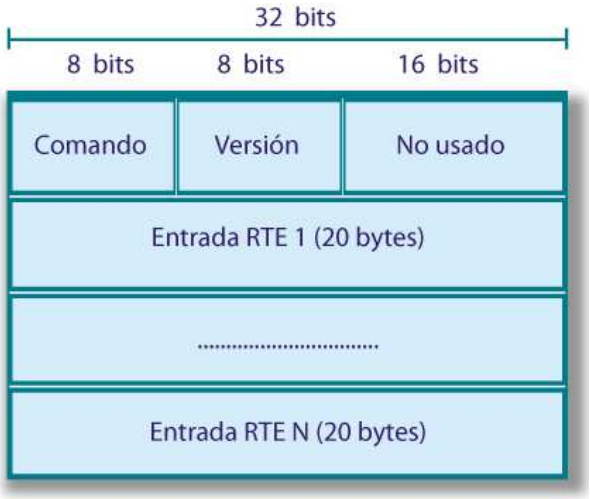

Figura 1- 3 Formato de Paquete RIPng

.

- Comando: Usado para especificar el propósito de este mensaje, estos paquetes son petición o respuesta, vistos ya en la versión 1 de RIP, este campo está formado por 8 bits.
- Versión: El número de versión es 1 para RIPng, está conformado por 8 bits.
- RTE - Tabla entrada de ruta: Para cada tipo de mensaje, se tiene una lista de tabla de entrada de ruta. En la que se especifican el prefijo destino, el número de bits significativos en el prefijo, y el costo de alcanzar ese destino (métrica). Está compuesto por 20 bytes.

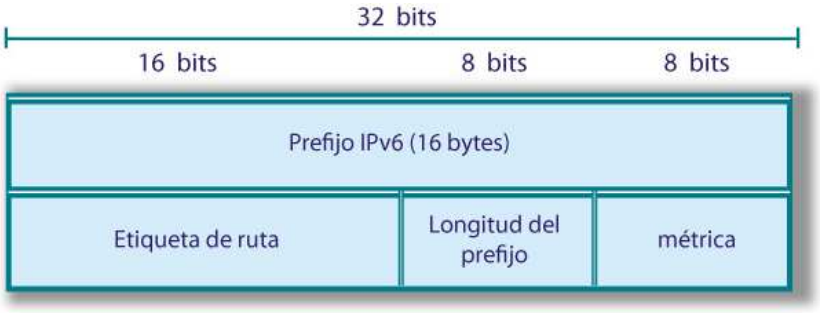

Figura 1- 4 Formato de campo RTE

- Prefijo IPv6: Está compuesto por 128 bits (16 octetos), especifica la dirección IPv6 destino.

- Etiqueta de ruta: Está asignado a una ruta, se lo implementa para proveer un método separación entre rutas RIP internas y externas.
- Longitud del prefijo: Indica la longitud en bits de la parte más significativa del prefijo, está formada por 8 bits.
- Métrica: Está constituida por 8 bits puede contener valores entre 1 y 15, permite especificar la métrica actual para alcanzar el destino. El valor 16 indica destino inalcanzable.

El tamaño máximo del datagrama está limitado por el MTU del medio sobre el cual el protocolo está siendo usado.

1.3.4 OSPF - OPEN SHORTEST PATH FIRST - ABRIR PRIMERO LA TRAYECTORIA MÁS CORTA

Creado para superar las limitaciones de RIP, es un protocolo de estado de enlace que funciona en base al algoritmo shortest path first (SPF).

1.3.4.1 Características

- Protocolo abierto, descrito en el RFC 1245.
- Soporta VLSM.
- No presenta limitaciones en el número de saltos.
- Tienen una convergencia rápida.
- Envía actualizaciones en base a direccionamiento multicast.
- Permite autenticación de rutas.
- Permite hacer una selección basada en el ancho de banda de las rutas.
- Soporta enrutamiento jerárquico.

1.3.4.2 Funcionamiento

Cada enrutador posee una base de datos de la topología de la red, envían mensajes a sus routers vecinos del estado de las rutas, esto se realiza

mediante inundación (flooding), estos mensajes son cortos, ya que no se envía toda la tabla de rutas únicamente el estado de los enlaces.

La red se divide en áreas, el área 0 es el área más importante representa el área de backbone a la cual por lo menos un router del resto de áreas debe estar conectado.

Los ruteadores contienen bases de datos de adyacencias que detallan la lista de los ruteadores vecinos, una base de datos topológica que contiene la lista de todas las rutas y una tabla de enrutamiento con las mejores rutas para alcanzar determinados destinos.

Permite balanceo de carga es decir si existen dos rutas con el mismo costo distribuye el tráfico equitativamente entre ellas.

1.3.4.3 **Protocolo Hello**

Es un protocolo auxiliar que utiliza OSPF para algunos propósitos:

1.3.4.3.1 Descubrimiento de Vecinos

En un segmento en común dos ruteadores son definidos como vecinos mediante el protocolo hello, para ello es necesario que parámetros como el área ID, los password de autenticación y los intervalos hello/dead sean valores fijados por ellos en mutuo acuerdo.

1.3.4.3.2 Elección del DR (Designated Router - Router designado) y BDR (Backup Designated Router - Router designado de Respaldo)

Se establecen prioridades en los ruteadores, el ruteador con mayor prioridad se elige como DR y el que tiene la siguiente prioridad se designa como BDR.

Dentro de una red LAN todos los ruteadores intercambian información con el DR y el BDR, si por algún motivo falla el DR el BDR es designado como nuevo DR y se procede a elegir un BDR de acuerdo a la prioridad.

Es importante mencionar que si dos ruteadores tienen la misma prioridad se procede a elegirlos mediante el mayor Router-ID es decir el ruteador que presente la dirección IP más grande configurada en las interfaces del enrutador.

1.3.4.4 **Formato Paquete OSPFv2**

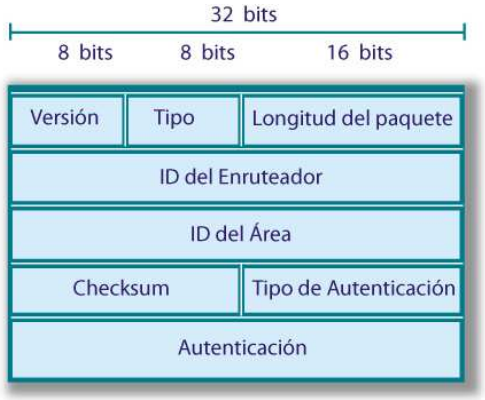

Figura 1- 5 Formato de Paquete OSPFv2

- Versión: Está compuesto por 8 bits, y permite identificar la versión de OSPF.
- Campo tipo: Está compuesto por 8 bits y permite definir qué tipo de paquetes van a ser enviados; estos pueden ser mensajes Hello, LSA[3], LSR[4], LSU[5], LSAcks[6], Database Description.
- Campo longitud de paquete: Consta de dos bytes y especifica la longitud total del paquete, incluyendo el encabezado.

---

[3] LSA (Link State Advertisement - Avisos de estado-enlace) Se notifica a la red los cambios en el estado de los enlaces de un ruteador mediante estos mensajes LSA.

[4] LSR (Link State Requirement) Requerimiento estado-enlace.

[5] LSU (Link State Update) Actualización estado-enlace.

[6] LSAcks (Link State Acknowledgement) paquetes que reconocen link-state updates.

- Campo identificador de ruteador "router ID": Está compuesto por cuatro bytes (32 bits) para identificar el origen del paquete.
- Campo identificador de área "área ID": Formado por cuatro bytes y permite identificar el área a la cual pertenece el paquete.
- Chequeo de errores "checksum": Tiene dos bytes de longitud y permite detectar errores, verifica toda la cabecera excepto los bits de autenticación.
- Tipo de autenticación: Compuesto de dos bytes y permite establecer el tipo de autenticación a usar.
- Campo autenticación: Está formado por 64 bits y contiene la información de autenticación.

1.3.4.5 **Paquete OSPFv3**

OSPFv3 proporciona la conexión con redes IPv6, está basado en OSPF versión 2, en la siguiente figura se muestra que su formato es similar a la versión anterior con pequeñas variaciones.

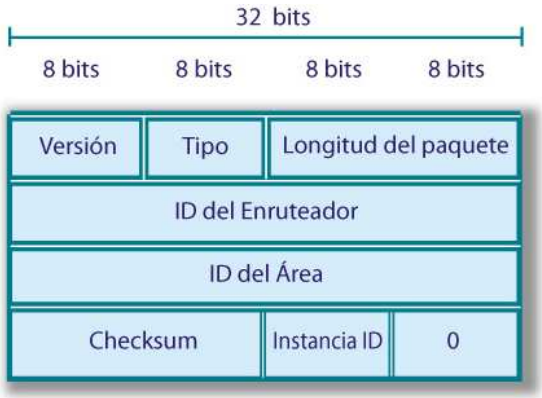

Figura 1- 6 Formato de Paquete OSPFv3

- Versión: Está compuesto por 8 bits, y permite identificar la versión de OSPF.
- Campo tipo: Está compuesto por 8 bits y permite definir qué tipo de paquetes van a ser enviados, estos pueden ser mensajes Hello, LSA, LSR, LSU, LSAcks, Database Description.

- Campo longitud de paquete: Consta de dos bytes y especifica la longitud total del paquete, incluyendo el encabezado.
- Campo identificador de ruteador "router ID": Está compuesto por cuatro bytes (32 bits) para identificar el origen del paquete.
- Campo identificador de área "área ID": Formado por cuatro bytes y permite identificar el área a la cual pertenece el paquete.
- Chequeo de errores "checksum": Tiene dos bytes de longitud y permite detectar errores, verifica toda la cabecera excepto los bits de autenticación.
- Campo Instancia ID: Es habilitado al usar enlaces simples.

1.3.5 BGP - BORDER GATEWAY PROTOCOL- PROTOCOLO PUERTA DE ENLACE DE BORDE

BGP utiliza el algoritmo vector distancia sin embargo difiere de RIP debido a que toma decisiones de enrutamiento basándose en políticas de la red, o reglas que utilizan varios atributos de ruta BGP, su finalidad es enrutar entre sistemas autónomos, es un protocolo de enrutamiento exterior, que sustituyó al EGP (Exterior Gateway Protocol) debido al crecimiento de las redes, y a su dificultad de detectar la presencia de bucles (creados por varios ruteadores al alcanzar otros sistemas autónomos al que ninguno está conectado).

Se han desarrollado cuatro versiones de BGP, las versiones uno y dos están obsoletas la versión 4 difiere de la tres por su soporte a CIDR (Classless Inter-Domain Routing - Encaminamiento Inter-Dominios sin Clases), debido a esta razón son incompatibles pero pueden negociarse el uso de ambas o de una en particular.

BGP ofrece fiabilidad en el transporte de datagramas, lo que elimina la necesidad de llevar a cabo la fragmentación, la retransmisión, el reconocimiento, y secuenciación.

La forma de configurar y delimitar la información que contiene e intercambia el protocolo BGP es creando lo que se conoce como Sistema Autónomo.

En un sistema autónomo se tiene sesiones internas (iBGP) y externas (eBGP), la información a enviarse en BGP se delimitan en estos sistemas autónomos, una sesión eBGP es cuando BGP está funcionando entre dos AS diferentes e iBGP cuando BGP está funcionado en el mismo AS.

1.3.5.1 **Mensajes BGP**

Los mensajes BGP son Open, Update, Notification, Keepalive, tienen una cabecera común de 19 octetos con los siguientes tres campos:

- Marcador: Reservado para autentificación, el emisor puede insertar un valor en este campo para permitir al receptor comprobar la veracidad del emisor.
- Longitud: Tamaño del mensaje en octetos.
- Tipo: Representa el tipo de mensaje, el cual puede ser Open, Update, Notification, y Keepalive.

Luego de abrir una conexión TCP un dispositivo envía un mensaje OPEN, para identificar el sistema autónomo al que pertenece el emisor y suministra la dirección IP del dispositivo de encaminamiento.

Los mensajes de notificación se envían cuando se detecta algún tipo de error, como por ejemplo errores en la cabecera del mensaje, error en el mensaje Open, Update, Keepalive o errores al expirar el tiempo de mantenimiento.

1.3.5.2 **Funcionamiento**

BGP realiza tres funciones principales, la adquisición de vecino debido a que enrutadores que pertenecen a un SA pueden necesitar intercambiar información entre ellos, detección de vecino alcanzable, aquí los dos ruteadores o vecinos tratan de no perder la conexión y por último la detección de red alcanzable.

En el procedimiento de detección de vecinos, un ruteador envía un mensaje OPEN a otro dispositivo para que acepte su petición, éste puede o no aceptar, una de las razones por la que podría negarse es por la sobresaturación en el tráfico que está manejando, si el dispositivo acepta la conexión envía un mensaje de KEEPALIVE, la dirección del receptor de este mensaje se establece previamente en la etapa de establecimiento de configuración del sistema.

En la detección de vecino alcanzable se intercambia frecuentemente mensajes de KEEPALIVE entre los dos vecinos, de esta forma se asegura que la relación sigue establecida.

En la detección de red alcanzable, los dispositivos de encaminamiento mantienen una base de datos en la que se especifica las redes alcanzables y la ruta preferida para llegar a esas redes.

Por último si ocurre algún cambio en esa base de datos, se envían mensajes de UPDATE por difusión a todos los dispositivos de encaminamiento que implementan BGP.

1.4 ANÁLISIS DE LA CAPA RED, IP INTERNET PROTOCOL - PROTOCOLO DE INTERNET

IP es un protocolo de capa red no confiable no orientado a conexión, se encarga de enviar paquetes de datos a través de la red para ello usa la técnica del mejor esfuerzo, establece una comunicación sin conexión, es importante mencionar que la corrección de errores en los paquetes enviados se realiza en las capas superiores ya que IP no implementa ningún mecanismo para corregirlos.

Los ruteadores son los encargados de encaminar los datagramas IP a través del Internet para ello revisan la dirección destino que se presenta en el datagramas IP, y escogen la mejor ruta para enviarlos.

Las tablas de enrutamiento contienen información de la interfaz más conveniente para transmitir el paquete IP, por esta razón cada ruteador examina su tabla de enrutamiento y determina el siguiente nodo por el que atravesará el datagrama IP.

1.4.1 PROTOCOLO IP VERSIÓN 4

IP versión 4 es un protocolo enrutable no orientado a conexión, es el principal protocolo de la arquitectura TCP/IP implementado a nivel de red. Fue desarrollado por el IETF Internet Engineering Task Force- Grupo de Trabajo en Ingeniería de Internet en Septiembre de 1981 y está oficialmente descrito en el RFC 791.

Su principal limitación es el espacio de direcciones, se pronostica que para el 2010 llegará a su límite, es por ello que su nueva versión está adquiriendo mayor fuerza en las comunicaciones a través del Internet.

1.4.1.1 **Características del Protocolo IPv4**

- Protocolo enrutable.
- No orientado a conexión.
- Técnica del mejor esfuerzo.
- No garantiza la entrega de datos.
- No garantiza la corrección de errores.
- Los paquetes pueden llegar duplicados o en desorden.
- Realiza el direccionamiento y la fragmentación de los datos.

Los protocolos de capas superiores proveen mecanismos para solucionar los problemas que presenta IP. Para que pueda ser identificada cada máquina es

necesario que tenga asignada una dirección[7] IP y así pueda intercambiar información.

Su función es tomar los datos (protocolo UDP o TCP) colocar una cabecera y encaminar este datagrama a través de la red, no realiza comprobación de integridad de la información.

1.4.1.2 **Formato del Protocolo IPv4**

La cabecea IP consta de 160 bits (20 Bytes), en la siguiente figura se muestran cada uno de sus campos:

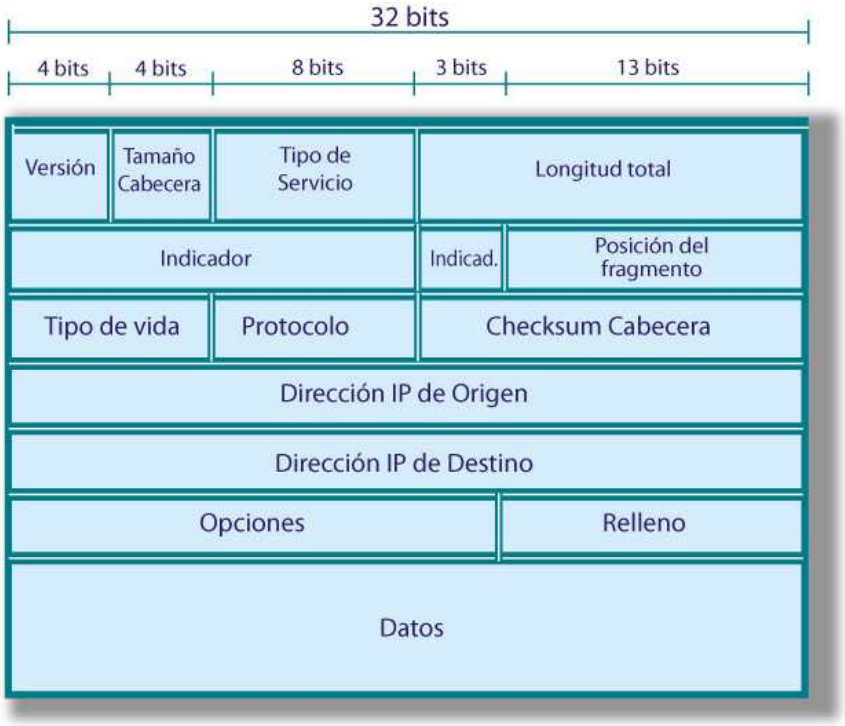

Figura 1- 7 Formato de Datagrama IPv4

[7] dirección es un indicativo, que permite identificar una máquina del resto de la red, es asignada a cada host

- Versión: Tiene un valor de 4 bits, permite identificar la versión del protocolo IP. Su valor para la versión actual es de 0100, 5 para la experimental y 6 representa IPng (IPv6).
- Tamaño de cabecera: Su tamaño es de 4 bits, identifica la longitud de la cabecera expresada en grupos de 32 bits. Su valor mínimo es de 5 para una cabecera correcta.
- Tipo de servicio: Permite indicar la prioridad de los datos enviados, su tamaño es de 8bits, 3 utilizados para identificar la prioridad (rango de 0-7, prioridad cero es considerada como baja y 7 representa la máxima prioridad que tiene un datagrama). Los 5 menos significativos indican las características del servicio:

  Bit 0: sin uso, debe permanecer en 0.

  Bit 1: 1 costo mínimo, 0 costo normal.

  Bit 2: 1 máxima fiabilidad, 0 fiabilidad normal.

  Bit 3: 1 máximo rendimiento, 0 rendimiento normal.

  Bit 4: 1 mínimo retardo, 0 retardo normal.

- Longitud total: Su tamaño es de 16 bits e indica el tamaño total del datagrama IP incluyendo la cabecera y los datos, está expresado en bytes. El tamaño máximo que puede tener el datagrama será de 65536 bytes.
- Identificador: Es único para cada datagrama, se lo utiliza cuando se fragmenta paquetes para poder identificarlos y ensamblarlos, los fragmentos de un datagrama tienen el mismo número de identificación. Está constituido por 16 bits.
- Indicadores o Banderas: Es utilizado para informar la fragmentación, está formado por 3 bits de los cuales solo se utilizan dos, la especificación de su uso es la siguiente:

  Bit 0: reservado siempre tiene el valor de cero

  Bit 1: (DF) 0 = Fragmentado, 1 = No fragmentado.

  Bit 2: (MF) 0 = Último fragmento, 1 = existen más fragmentos.

- Posición del fragmento: Permite identificar la posición que ocupa el fragmento en el paquete original. Está formado por 13 bits y ayuda al ensamblaje correcto de los fragmentos enviados, este valor se pone en cero si el paquete no ha sido fragmentado. Se mide en bytes lo cual determina que cada fragmento debe ser un número entero de bytes. El tamaño mínimo de un fragmento es de 8 bytes de datos más encabezado.
- Tiempo de vida o TTL: Este campo es seteado por quien envía el datagrama, está constituido por 8 bits, permite definir el tiempo máximo que un datagrama puede estar en la red, es decrementado en un valor de uno por cada ruteador que atraviesa el datagrama IP. Cuando este campo tiene un valor de cero, se descarta el datagrama y se genera un mensaje ICMP[8] que indica tiempo excedido al host que originó el paquete.
- Protocolo: Indica el protocolo utilizado en el campo datos del datagrama, está formado por 8 bits.
- Header checksum: Es un mecanismo de detección de errores únicamente para el encabezado IP. Se lo determina dividiendo la cabecera IP en palabras de 16 bits, sumando estas palabras en complemento de uno y sacando el complemento de uno del resultado. Está conformado por 16 bits. En cada nodo el checksum presenta un valor diferente debido a la variación de los parámetros por ejemplo el tiempo de vida del datagrama.
- Dirección origen: Está conformada por 32 bits y contiene la dirección IP del origen.
- Dirección destino: Está conformada por 32 bits y contiene la dirección IP del destino.

[8] ICMP (Internet Control Message Protocol) Protocolo de Control de Mensajes de Internet. Permite identificar y notifica errores del protocolo IP.

- Opciones IP: Permite que en IP se implementen facilidades para depuración, medición y seguridad, no es un campo obligatorio.
- Relleno: Se lo utiliza para garantizar que el encabezado IP presente un número entero de palabras de 32 bits, si existe un campo opciones IP y no completa un múltiplo de 32 bits se lo completa.

1.4.1.3 **Direccionamiento IPv4**

Las direcciones IP versión 4 son de 32 bits, permiten identificar de manera lógica y jerárquica a una interfaz de un dispositivo para que pueda acceder al Internet. Se los representa en notación decimal.

Número de direcciones IPv4 = 2^{32} = 4.294.967.296

Estos 32 bits equivalen a 8 octetos cuyos valores van de 0 a 255, cada octeto está separado por un punto.

El direccionamiento está basado en clases, la ICANN - Internet Corporation for Assigned Names and Numbers - Corporación de Internet para la Asignación de Nombres y Números se encarga de la asignación de direcciones, la clase A fue utilizada para grandes empresas actualmente este organismo las reserva para los gobiernos de todo el mundo, la clase B se asigna a medianas empresas y la clase C para el resto de solicitantes.

1.4.1.3.1 Notación Decimal Separada por Puntos

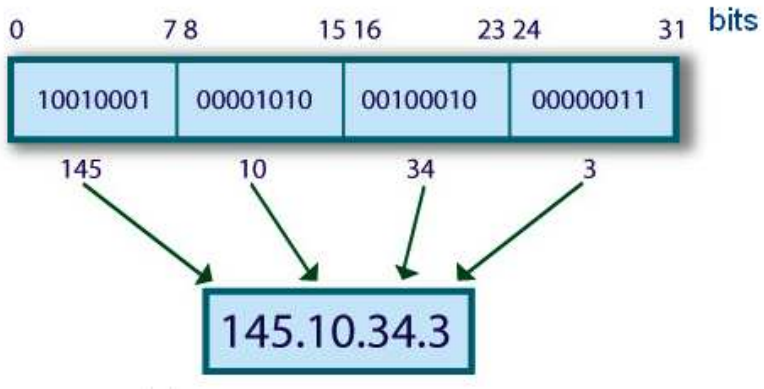

Figura 1- 8 Ejemplo de Notación Decimal

Cada dirección IP está formada por dos partes, los primeros bits son destinados a identificar la red y la otra parte formada por los últimos bits sirven para identificar los host.

En la clase A se destina un octeto para las direcciones de red y los tres últimos octetos para la dirección de host de esta manera la cantidad máxima de hosts es $2^{24} – 2$ (las direcciones reservadas de broadcast [últimos octetos a 255 unos lógicos] y de red [últimos octetos a 0]), es decir, 16 777 214 hosts).

En una red clase B, se asignan los dos primeros octetos para identificar la red y los dos octetos finales para los hosts, de esta manera existen 2^{14} redes clase B o 15384, y la cantidad máxima de hosts es 2^{16} - 2, o 65 534 hosts.

En una red de clase C, se asignan los tres primeros octetos para identificar la red, reservando el octeto final para que sea asignado a los hosts, de este modo existen 2^{21} redes clase C o 2.097.157 y la cantidad máxima de hosts es 2^{8} - 2, o 254 hosts.

La clase D es considerada como las redes cuyo rango es 11100000 a 11101111, estas redes no representan una máquina sino una colección que forma parte de un grupo multicast, comprende las direcciones de red desde la 224.0.0.0 hasta la 239.255.255.255.

Una dirección multicast es una dirección exclusiva de red que dirige los paquetes con esa dirección destino hacia grupos predefinidos de direcciones IP. Por lo tanto, una sola estación puede transmitir de forma simultánea una sola corriente de datos a múltiples receptores. [8]

La clase E está compuesta por las redes comprendidas desde la 240.0.0.0 hasta la 247.255.255.255, esta clase es reservada para un uso futuro.

La IETF Internet Engineering Task Force - Fuerza de tareas de ingeniería de Internet ha reservado estas direcciones para su propia investigación. Por lo tanto, no se han emitido direcciones Clase E para ser utilizadas en Internet. Los primeros cuatro bits de una dirección Clase E siempre son 1s. Por lo tanto, el

rango del primer octeto para las direcciones Clase E es 11110000 a 11111111. [8]

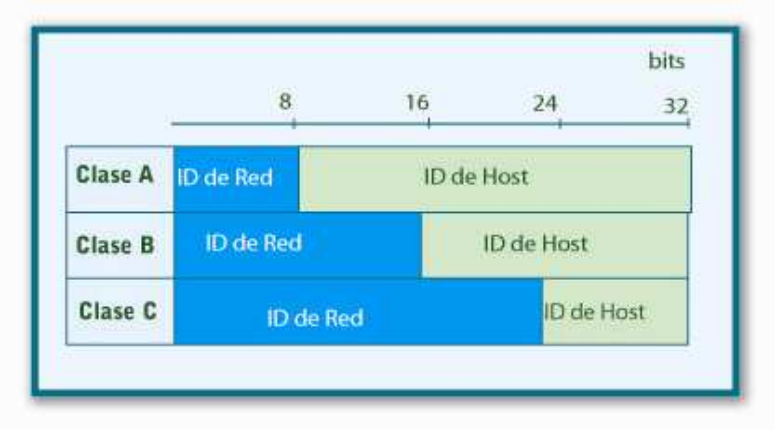

Figura 1- 9 Clases en las Direcciones IP

Rangos de direcciones para cada una de las clases.

Clase	Dirección	
	Desde	Hasta
A	1.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Tabla 1- 2 Formatos de Dirección

Existen dentro de cada clase de direcciones IP direcciones no asignadas estas se denominan privadas, no válidas no ruteables desde el Internet, son utilizadas por redes que no tienen suficientes direcciones para cada uno de sus hosts es por ello que utilizan un servidor de traducción de direcciones de red (NAT) para lograr conectarse al Internet. Las direcciones privadas se utilizan para el direccionamiento de máquinas dentro de una red privada, contribuyen a

solucionar el problema de agotamiento de direcciones, en la tabla 1-3 se muestran sus rangos con sus respectivas máscaras.

- Clase A: 10.0.0.0 a 10.255.255.255 (8 bits red, 24 bits hosts).
- Clase B: 172.16.0.0 a 172.31.255.255 (16 bits red, 16 bits hosts).
- Clase C: 192.168.0.0 a 192.168.255.255 (24 bits red, 8 bits hosts).

Rango	Máscara	CIDR
10.0.0.0 - 10.255.255.255	255.0.0.0	/8
172.16.0.0 - 172.31.255.255	255.240.0.0	/12
192.168.0.0 - 192.168.255.255	255.255.0.0	/16

Tabla 1- 3 Direcciones IP privadas

1.4.1.3.2 Máscaras de Subred

Es importante recordar que una Dirección IP va acompañada de la máscara de subred. La máscara de la subred indica qué bits son números de subred y cuales pertenecen al número de hosts.

Se utiliza separar diferentes redes, con el fin de facilitar la administración y reducir tráfico inútil.

Las máscaras de subred son fundamentales para las comunicaciones sobre una red IP, con éstas se logra dividir grandes redes en redes menores. En la tabla 1-4 se especifica cómo se obtienen las máscaras para cada clase de direcciones IP.

Clase	Formato Binario	Formato decimal	Número de Hots
A	11111111.00000000.00000000.000000000	255.0.0.0	16777214
B	11111111.11111111.00000000.000000000	255.255.0.0	65534
C	11111111.11111111.00000000.000000000	255.255.255.0	254

Tabla 1- 4 Máscaras de Subred de cada Clase

Dada una dirección IP y la máscara de subred al aplicar AND lógico se puede obtener la red a la que pertenece determinado host.

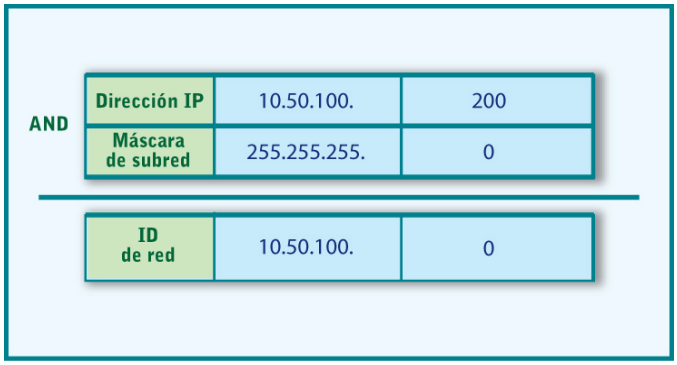

Figura 1- 10 Ejemplo de lógico para máscaras de subred

.

Al asignar un ID de red se tiene un número fijo de host y un identificador de red único, de esta manera una organización sólo puede tener una red y un número asignado de hosts conectándose a la misma, si se presenta el caso de tener un número grande de hosts con una red única, ésta no podrá funcionar eficazmente. Para solucionar este problema, se introdujo el concepto de subredes.

1.4.1.3.3 Subredes

Una subred es un conjunto de dispositivos de red. Gracias a la segmentación de redes se logra acomodar la carga, la información se transmitirá de una forma más clara (sin colisiones) y por lo tanto más rápidamente. Las subredes permiten obtener ID de red de menor tamaño dividiendo el único ID de red de una clase.

La asignación de máscaras definida en la tabla 1.4 provoca un gran desperdicio de direcciones por ejemplo si necesitamos conectar 7 hosts usando una dirección clase C con máscara 255.255.255.0 se desperdiciarán 247 direcciones, por ello se utiliza VLSM (Variable-Length Subnet Masking - Máscara de subred de longitud variable), para hacer posible la asignación de prefijos de longitud arbitraria.

Para crear una subred, se debe tener en cuenta lo siguiente:

- Definir la cantidad de bits que se usarán para la subred, esto depende del número de hosts que se necesita para cada subred.
- Calcular a máscara a configurar en los dispositivos.

Por ejemplo si tenemos la dirección 193.1.1.0 y necesitamos definir 6 subredes, con 25 hosts por cada subred.

La dirección 192.1.1.0 es una dirección clase C.

193 . 1 . 1. 0 0 0 | 0 0 0 0 0

Subred hosts

6 subredes + 2 = 8 $8 = 2^3$ → bits para subredes

8 – 3 = 5 bits para hosts

2^5 – 2 = 30 hosts / subred

Dirección de Subred	Prefijo de red	Bits para Hosts (sufijo para hosts)	Descripción
193.1.1.	0 0 0	0 0 0 0 0	Subred con dirección igual a la de red y no es utilizada
193.1.1.	0 0 1	0 0 0 0 0	1 Subred
193.1.1.	0 1 0	0 0 0 0 0	2 Subred
193.1.1.	0 1 1	0 0 0 0 0	3 Subred
193.1.1.	1 0 0	0 0 0 0 0	4 Subred
193.1.1.	1 0 1	0 0 0 0 0	5 Subred
193.1.1.	1 1 0	0 0 0 0 0	6 Subred
193.1.1.	1 1 1	0 0 0 0 0	No la usamos

Tabla 1- 5 Ejemplo subneteo

Para conocer las direcciones IP de la primera red:

193 . 1 . 1 .	**0 0 1**	**0 0 0 0 1**	**193.1.1.33**	**Primera dirección IP**
193 . 1 . 1 .	**0 0 1**	**1 1 1 1 0**	**193.1.1.62**	**Última dirección IP**
193 . 1 . 1 .	**0 0 1**	**1 1 1 1 1**	**193.1.1.63**	**Dirección de broadcast**

Tabla 1- 6 Ejemplo subneteo 2

El esquema de direccionamiento correspondiente para cada subred es el siguiente:

Dirección de Subred	1º dirección IP valida	Última dirección IP valida	Dirección de Broadcast	Máscara de Subred
193.1.1.32	193.1.1.33	193.1.1.62	193.1.1.63	255.255.255.224
193.1.1.64	193.1.1.65	193.1.1.94	193.1.1.95	255.255.255.224
193.1.1.96	193.1.1.97	193.1.1.126	193.1.1.127	255.255.255.224
193.1.1.128	193.1.1.129	193.1.1.158	193.1.1.159	255.255.255.224
193.1.1.160	193.1.1.161	193.1.1.190	193.1.1.191	255.255.255.224
193.1.1.192	193.1.1.193	193.1.1.222	193.1.1.223	255.255.255.224

Tabla 1- 7 Ejemplo esquema de direccionamiento

Para el ruteo se debe separar el prefijo de red del sufijo de host, es decir se rutea en base al prefijo. Sólo cuando ya se está en la red correcta se usa el sufijo de host para encontrar la máquina destino en la red local.

Una dirección CIDR implementa la técnica VLSM para especificar prefijos de red, una dirección CIDR se escribe con un sufijo que indica el número de bits de longitud de prefijo, de la siguiente manera: 192.168.0.0/16 los primeros cuatro números decimales se interpretan como una dirección IPv4, y el número a continuación de la barra es la longitud de prefijo.

La forma de determinar la máscara es colocando unos, en tantos bits como marque el prefijo comenzando desde la izquierda, el resto colocamos en ceros y separamos estos 32 bits en grupos de 8 bits.

De acuerdo a las necesidades de cada organización se procede a segmentar la red, para ello se utiliza CIDR que implementa VLSM para asignar direcciones IP a subredes, el proceso de asignación es recursivo y llevado a cabo en cualquier bit de los 32 que componen la dirección IP.

1.4.2 PROTOCOLO IP VERSIÓN 6

IPv6 - Protocolo de Internet versión 6 fue diseñado por Steve Deering de Xerox PARC y Craig Mudge, para suplir el limitante del número de direcciones proporcionadas por el protocolo de Internet versión 4 y para aumentar nuevas funciones en el estándar IPv6.

1.4.2.1 **Direccionamiento IPv6**

Ipv6 cambia el tamaño de dirección IP de 32 bits (IPv4) a 128 bits, para garantizar un mayor número de direcciones IP, lo que hace posible prescindir el uso de NAT Network Address Translator - Traductor de direcciones de red, esto conlleva a que empresas que poseen un número limitado de direcciones IP públicas se vean en la necesidad de usar direcciones privadas que apunten a una única IP pública para enviar la información de su red interna a la red externa, además el incremento de bits en el tamaño de dirección IP garantiza direccionamiento flexible y perdurable.

Número de direcciones IPv6 = $2^{128} \approx 3.4 \times 10^{38} = 16^{32}$

Es decir los 128 bits equivalen a 32 dígitos hexadecimales que pueden tomar 16 combinaciones posibles.

1.4.2.2 **Notación de direcciones IPv6**

Los 32 dígitos hexadecimales que componen el tamaño de dirección IP son representados por 8 grupos de 4 dígitos cada uno, de la siguiente manera:

1080:2ab8:05c2:20d3:7035:48db:0e35:9101

Las direcciones IPv6 pueden ser comprimidas si uno o varios grupos de dígitos presentan en su campo un valor de “0000”.

1080:2ab8:05c2:20d3:0000:48db:0e35:9101

Dirección Comprimida:

1080:2ab8:05c2:20d3::48db:0e35:9101

La compresión de más grupos de campos cuyo valor es “0000” se representa de la siguiente manera:

1080:2ab8:0000:0000:0000:0000:0e35:9101

1080:2ab8:0000:0000:0000::0e35:9101

1080:2ab8:0:0:0:0:0e35:9101

1080:2ab8:0::0:0e35:9101

1080:2ab8::0e35:9101

Si un grupo de dígitos está precedido por un cero este puede ser borrado, cabe señalar que el símbolo :: solo se presenta una única vez en una dirección IPv6, y será usado para representar uno o más grupos de 16 bits de ceros.

1.4.2.2.1 Tipos de direcciones

Unicast: Utiliza 64 bits menos significativos de la dirección Ipv6. Se la implementa en una interfaz cuando envía la información a un único destino.

Multicast: Es una dirección para un conjunto de interfaces y se entrega a todos ellos.

Anycast: Se la asigna para un conjunto de interfaces pero únicamente se entrega al más cercano.

En el IPv6 no existen direcciones broadcast, su funcionalidad ha sido mejorada por las direcciones multicast.

Las direcciones IPv4 de tipo broadcast (normalmente xxx.xxx.xxx.255) se expresan en IPv6 mediante direcciones multicast.

1.4.2.2.2 Direcciones Especiales

Dentro de Ipv6 existen direcciones especiales como son por ejemplo: la dirección no especificada (0:0:0:0:0:0:0:0 ó ::), esta se utiliza para indicar ausencia de dirección.

Otro ejemplo, es la dirección de loopback (0:0:0:0:0:0:0:1 ó ::1), la cual permite que un nodo no se envíe direcciones a sí mismo. Equivale a la dirección IPv4 de loopback: 127.0.0.1.

Existen direcciones compatibles, esto con el fin de facilitar la migración de IPv4 a IPv6 y la coexistencia de tipos de host con direcciones IPv4 e IPv6, este tipo de direcciones son las siguientes:

- ::<Dirección de IPV4> se utiliza para los túneles dinámicos de IPV6 sobre IPV4.
- ::FFFF: <Dirección de IPV4> se utiliza para los túneles dinámicos de IPV4 sobre IPV6.

Un ejemplo de ::<Dirección de IPV4> puede escribirse de la siguiente manera:

::ffff:192.168.89.9

La dirección: 192.168.89.9 = 11000000.10101000.01011001.00001001 = c0a85909

El resultado es el siguiente:

::ffff:c0a8:5909

Se denomina dirección IPv4 mapeada al formato ::ffff:1.2.3.4 y dirección IPv4 compatible al formato ::1.2.3.4

La dirección IPv4 compatible casi no está siendo utilizada en la práctica, aunque los estándares no la han declarado obsoleta.

1.4.2.2.3 Prefijos de direcciones IPv6

Los prefijos de las direcciones IPv6 asignadas nos permiten identificar a que tipo pertenecen.

Éstos prefijos se asignan de manera similar a las direcciones IPv4, igual siguen el esquema CIDR: dirección IPv6/ longitud del prefijo.

Por ejemplo, para la dirección IPv6 12AB: 0:0: CD30:: / 60, su prefijo en hexadecimal y en binario.

Prefijo:

12AB00000000CD3 (hexadecimal):

0001 0010 1010 1011 0000 0000 0000 0000 0000 0000 0000 0000 1100 1101 0011 (binario)

Se puede expresar:

12AB: 0000:0000: CD30: 0000:0000:0000:0000 / 60

12AB:: CD30: 0:0:0:0 / 60

12AB: 0:0: CD30:: / 60

A continuación la tabla 1-5 que detalla la asignación de los prefijos en binario.

ESPACIO	PREFIJO
0000.0000	Reservada (aquí se incluyen las de IPv4)
0000.0001	No asignada
0000.001	Reservadas para NSAP
0000.010	Reservada para IPX
0000.011	No asignada
0000.1	No asignada
0001	No asignada
001	No asignada
010	Direcciones basadas en el proveedor
011	No asignada
100	Direcciones basadas geográficamente
101	No asignada
110	No asignada
1110	No asignada
1111.0	No asignada
1111.10	No asignada
1111.110	No asignada
1111.1110.0	No asignada
1111.1110.10	Direcciones de uso local
1111.1110.11	Direcciones de sitio local
1111.1111	Direcciones de multicast

Tabla 1- 8 Prefijos de Direcciones IPv6

1.4.2.3 **Paquetes IPv6**

El paquete Ipv6 consta de una cabecera que corresponde a 40 bytes y el campo de datos que alcanza un tamaño en modo normal de 64 kB en paquetes normales, o mayor a este construyendo tramas con la opción "jumbo payload", es decir usando la cabecera de extensión de salto a salto.

La cabecera del protocolo de Internet versión 6 consta de los siguientes campos:

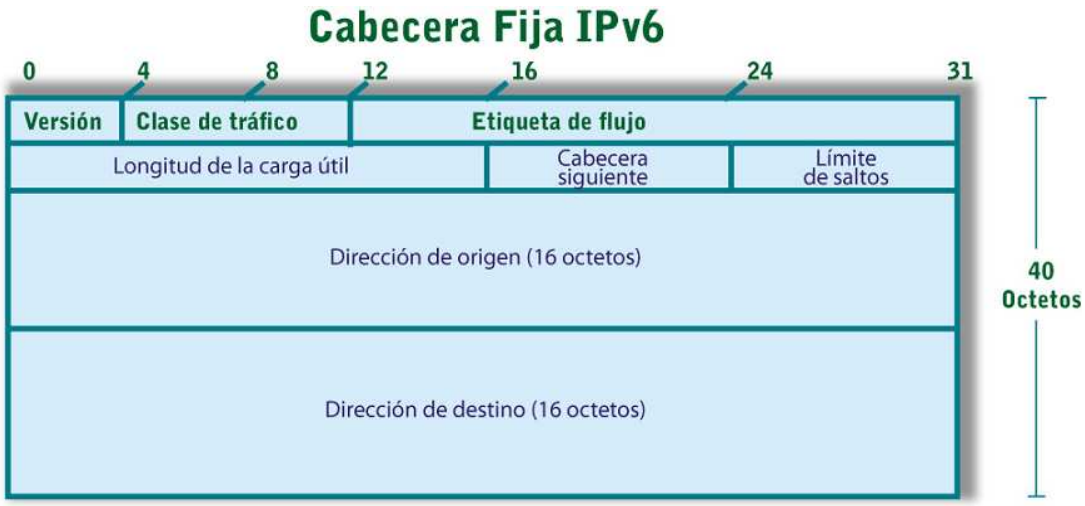

Figura 1- 11 Cabecera IPv6

- Versión: Número = 6 de versión del protocolo Internet de 4 bits.
- Clase de tráfico: Campo clase de tráfico de 8 bits.
- Etiqueta de flujo: Etiqueta de flujo de 20 bits.
- Longitud de la carga útil: Entero sin signo de 16 bits. Longitud de la carga útil IPv6, es decir, el resto del paquete que sigue a esta cabecera IPv6, en octetos. (Notar que cualquiera de las cabeceras de extensión presente es considerada parte de la carga útil, es decir, incluida en el conteo de la longitud).

- Cabecera siguiente: Selector de 8 bits. Identifica el tipo de cabecera que sigue inmediatamente a la cabecera IPv6. Utiliza los mismos valores que el campo protocolo del IPv4.
- Límite de saltos: Entero sin signo de 8 bits. Decrementado en 1 por cada nodo que reenvía el paquete. Se descarta el paquete si el límite de saltos es decrementado hasta cero.
- Dirección origen: Dirección de 128 bits del originador del paquete.
- Dirección destino: Dirección de 128 bits del recipiente pretendido del paquete (posiblemente no el último recipiente, si está presente una cabecera Enrutamiento).[5]

1.4.2.4 **Cabeceras de Extensión**

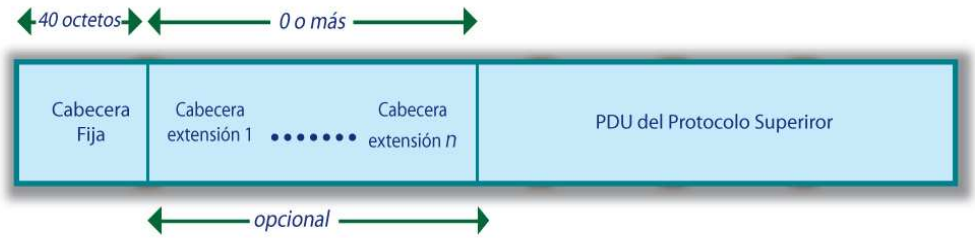

Figura 1- 12 Estructura del Datagrama IPv6

Los campos cabeceras extendidas son cabeceras que proporcionan una información adicional a la suministrada por la cabecera principal, éstas pueden utilizarse o no.

El número de cabeceras de extensión varía entre 0 y 8, se ubican entre la cabecera fija y la carga útil, especificamos la continuación de una cabecera de extensión en el campo siguiente de la cabecera anterior. Son enteros múltiplos de 8 octetos.

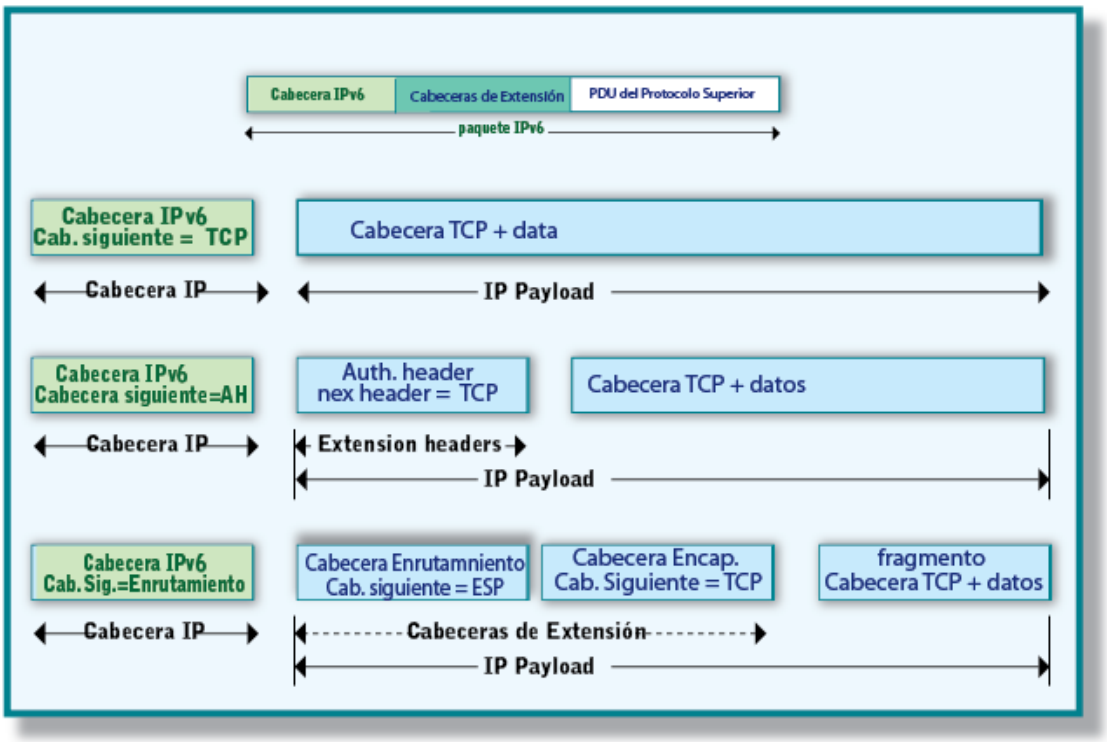

Figura 1- 13 Cabeceras de extensión

En el campo cabecera siguiente se coloca el tipo de cabecera que sigue o el identificador de protocolo de nivel superior, es necesario mencionar que estas cabeceras opcionales no se procesan en la ruta, únicamente en el nodo destino, a excepción de la cabecera hop by hop la cual contiene información importante de los saltos que realiza el paquete en la ruta hasta alcanzar su destino.

Se utiliza mensajes ICMP Internet Control Message Protocol - Protocolo de Mensajes de Control de Internet si un nodo al analizar la cabecera siguiente no encuentra el valor de la cabecera que continúa, o si se encuentra un valor de cero en el campo cabecera siguiente a excepción de la cabecera IPv6.

1.4.2.4.1 Tipos de cabeceras de extensión

- Cabecera Opciones de Salto a Salto: Se determina su presencia colocando un valor de 0 en la cabecera siguiente de la cabecera IPv6,

transporta datos que son examinados por cada nodo de la ruta que atraviesa el paquete.

- Cabecera Opciones de Destino: Se puede presentar dos veces antes de la cabecera de enrutamiento y después de cabecera seguridad del encapsulado de la carga útil, su código es 60, y únicamente su información es analizada por el o los nodos destino.
- Cabecera Enrutamiento: Se especifica los nodos que el paquete va a visitar para alcanzar su destino. Su código es 43.
- Cabecera Fragmento: Se utiliza si se envía un paquete con un MTU - Unidad Máxima de Transferencia mayor al especificado, cabe recalcar que esta opción es únicamente válida en el nodo origen al contrario que en IPv4 ya que la fragmentación del paquete se realiza en cada nodo. Esta opción nos proporciona flexibilidad en el envío de paquetes logrando transmitirlos superando el MTU de la ruta especificada. Cada fragmento del paquete tiene identificadores diferentes y direcciones origen y destino similares.
- Cabecera Autenticación: Se implementa para proveer integridad[9] y autenticidad[10] en el envío de información.
- Cabecera Seguridad del Encapsulado de la Carga Útil: Se implementa para proveer integridad, autenticidad y confidencialidad[11] en el envío de información.
- Cabecera No Hay Siguiente: Su código es 59 y representa que no existe ningún valor siguiendo la cabecera, si existen octetos luego de encontrar un valor de 59 en la cabecera siguiente de la cabecera IPv6 o de las de extensión simplemente son ignorados.

---

[9] Integridad se dice que la información es íntegra si no ha sufrido alteraciones

[10] Autenticidad permite validar si el emisor es quien dice ser, además permite garantizar que la información recibida es igual a la enviada.

[11] Confidencialidad, garantiza que personas no autorizadas tengan acceso a la información.

El orden de las cabeceras del paquete IPv6 es el siguiente:

- Cabecera IPv6.
- Cabecera Opciones de Salto a Salto.
- Cabecera Opciones de Destino.
- Cabecera Enrutamiento.
- Cabecera Fragmento.
- Cabecera Autenticación.
- Cabecera Seguridad del Encapsulado de la Carga Útil.
- Cabecera Opciones de Destino.
- Cabecera de Capa Superior.

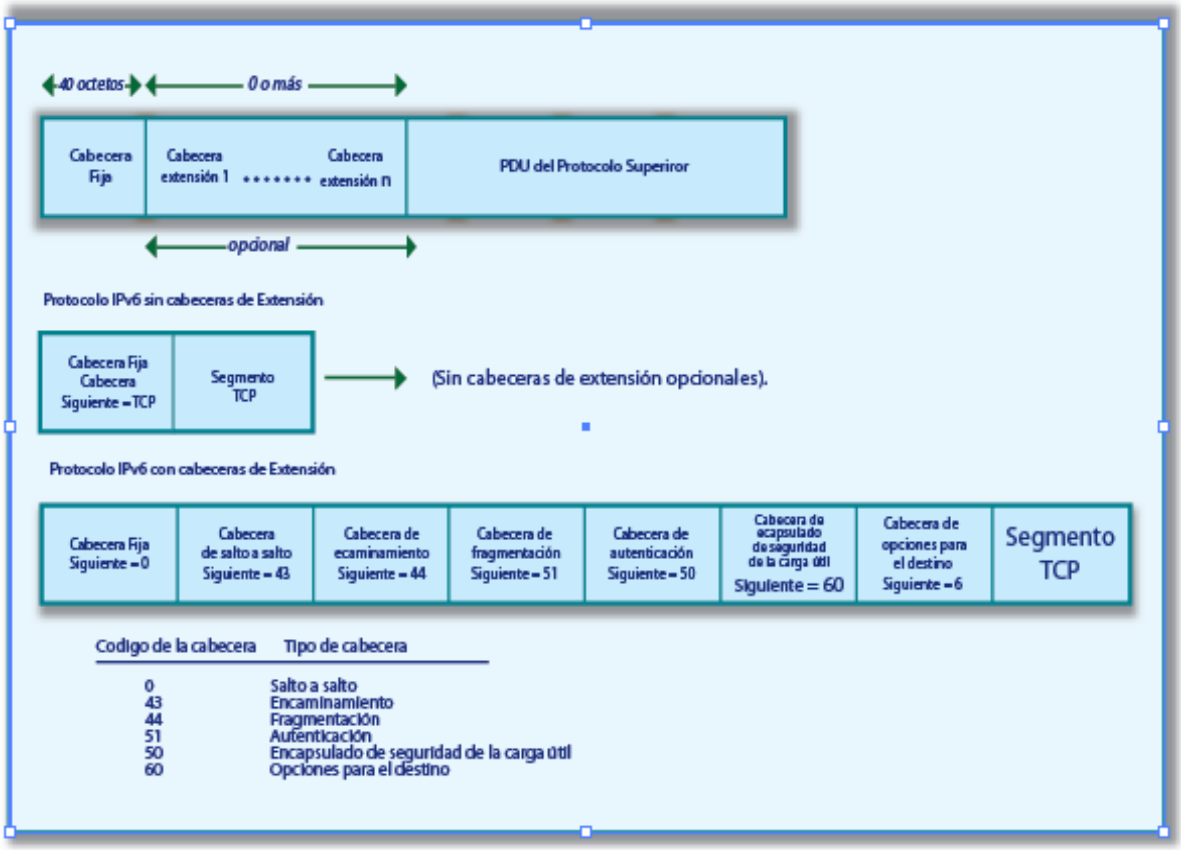

Figura 1- 14 Secuencia de las Cabeceras de Extensión

La primera cabecera opciones de destino su información es procesada por el primer destino que presenta la cabecera IPv6 en su campo dirección destino más las siguientes direcciones destino que se indican en el enrutamiento, la

cabecera *Opciones de destino* ubicada inmediatamente antes de la cabecera de *Capa superior* especifica la información procesada únicamente por el nodo final al que llega el paquete IPv6.

1.4.2.5 **Tamaño del paquete IPv6**

Los enlaces de Internet tienen MTU específico para el envío de paquetes, es necesario que el MTU para enviar paquetes IPv6 sea de 1280 octetos o mayor, en ciertos enlaces tales como PPP se configura el enlace para obtener un MTU de 1280 octetos sin embargo se recomienda configurarlos a 1500 octetos, de lo contrario se debe fragmentar el paquete IPv6 para lograr su transmisión, esta fragmentación se realiza en la capa de enlace.

En el RFC2460 del protocolo IPv6 se recomienda la utilización de la opción del descubrimiento del MTU de la ruta implementada en los nodos IPv6 para tomar ventaja en el envío de paquetes sin embargo también se señala que en implementaciones mínimas IPv6 se omite esta opción y se envía paquetes menores a 1280 octetos.

Se debe señalar que si se envían paquetes de tamaño mayor al MTU de la ruta, lógicamente fragmentados se debe tener la seguridad que el nodo destino es capaz de reensamblarlos y que tiene la capacidad de aceptar paquetes de un tamaño tan grande como 1500 octetos.

Al enviar paquetes IPv6 a un destino IPv4 el nodo que origina el paquete debe incluir la cabecera fragmento para dividir el paquete y así el enrutador traductor que realiza esta conversión IPv6 a IPv4 incluirá identificadores apropiados para cada fragmento IPv4 obtenido.

1.4.2.6 **DNS en Ipv6**

Un DNS Domain Name System - Sistema de Nombres de Dominio, permite una búsqueda directa correlacionando nombres de hosts con direcciones IPv4, éste

tipo de consulta realizada por las aplicaciones retornan direcciones IPv4 de 32 bits.

Para poder almacenar direcciones IPv6 se definió el RFC 3596, en el que se utiliza un nuevo tipo de registro AAAA desarrollado para almacenar direcciones IPv6 de 128 bits.

Un ejemplo de expediente AAAA es:

moon.universe.com EN AAAA 4321:0:1:2:3:4:567:89ab

Aquí se distingue la dirección IPv6 asignada a un nombre de quipo.

1.4.3 COMPARACIÓN ENTRE LOS PROTOCOLOS IPV4 E IPV6

IPv4	IPv6
Espacio de direcciones de 32 bits, es decir 2^{32} ~ $4.2x10^{9}$ direcciones IP posibles.	Espacio de direcciones de 128 bits, es decir 2^{128} ~ $3.4x10^{34}$ direcciones IP posibles.
Configuración manual o dinámica (DHCP).	Configuración “Plug & Play”, manual o dinámica (DHCPv6).
Direcciones de tipo unicast, multicast y broadcast.	Direcciones de tipo unicast, multicast y anycast.
Políticas de calidad de servicio se realizan a través del campo Tipo de Servicio (ToS) del paquete IP.	Políticas de calidad de servicio se realizan a través de los campos Etiqueta de Flujo y Clase de Tráfico.
Seguridad es algo opcional, a través del parche IPSec.	Seguridad extremo-a-extremo implementada en forma nativa.
Protocolo no escalable.	Protocolo escalable.
Movilidad mucho más difícil de implementar. En IPv4 no existe este concepto.	Movilidad más rápida, ya que no se pierde tiempo en el traslado de nodos.
	Tablas de enrutamiento mucho menores, ya que utiliza direccionamiento jerárquico.
	Extensibilidad, usa cabeceras auxiliares.

El proceso de fragmentación en Ipv4 genera más retardo y fluctuación a la transmisión de la videoconferencia.	
Como el proceso de fragmentación se realiza en los equipos intermedios, permite que la tasa de errores se mantenga baja cuando la topología del trayecto cambie.	El proceso de fragmentación de Ipv6 que se realiza por defecto en el origen es eficiente cuando la topología del trayecto se mantiene estable.
	Por sus características nativas de proveer calidad de servicio y seguridad en el transporte de información, IPv6 genera una arquitectura ideal para aplicaciones de tiempo real.
	Los encaminadores deben ser usados preferentemente para encaminar la información y no al análisis de ésta; IPv6 minimiza el uso de recursos de los encaminadores.
Utiliza registros de recurso (A) de dirección de host en el sistema de nombres de dominio (DNS) para correlacionar nombres de host con direcciones IPv4.	Utiliza registros de recurso (AAA) de dirección de host en el sistema de nombres de dominio (DNS) para correlacionar nombres de host con direcciones IPv6.
Utiliza el campo Time to Live para percatarse del tiempo que el paquete está viajando por la red**.**	En IPv6 se ha cambiado el campo Time to Live de IPv4 por el campo Hop Limit, para identificar el número de nodos o enrutadores por los que pasa el paquete hasta llegar al destino final.
En Ipv4 se tiene que incluir un tamaño fijo del encabezado se utilice o no en su totalidad.	Ipv6 presenta ventaja en cuanto al tamaño de bytes que ocupan los encabezados, los cuales son variables.

Tabla 1- 9 Comparación Datagrama IPv4 vs IPv6

1.5 CALIDAD DE SERVICIO

En redes que operan en base al mejor esfuerzo como lo es IP, todos los paquetes son entregados por igual y al existir congestión en la red cualquiera de ellos tienen la misma probabilidad de ser descartado. El concepto de calidad de servicio surge por la necesidad de que cierta información como aplicaciones en tiempo real (voz, video), lleguen sin retardo ni pérdida de información, por ello es deseable manejar prioridades de tráfico para la información más sensible a ser enviada.

En una red con múltiples usuarios, al compartir la conexión a Internet y tratar de bajar ciertos archivos, algunos de los usuarios bajan notablemente su ancho de banda hasta casi perderlo, por otra parte al congestionar uno de los canales (de subida o bajada) el otro canal disminuye su ancho de banda.

Con QoS se pretende priorizar la información más relevante, haciendo un uso eficiente de los recursos de la red al existir congestión, de este modo se procura que usuarios que acaparan la conexión a Internet y que dejan al resto de usuarios con un mínimo ancho de banda y tiempos de respuesta demasiado elevados no lo hagan.

Si se congestiona un canal debido a que no se pudo enviar todos los paquetes a su destino, además la imposibilidad de almacenamiento ilimitado de un ruteador ya que cuenta con buffers internos limitados provoca una pérdida de ancho de banda del otro canal.

1.5.1 MODELOS DE CALIDAD DE SERVICIO

Existen tres modelos en los que se divide el despliegue de calidad de servicio:

1.5.1.1 **Servicio de Mejor Esfuerzo**

En este servicio la red realiza el mejor esfuerzo para intentar entregar el paquete a su destino, sin embargo no existe ninguna garantía de su llegada,

por esta razón no es apropiado para aplicaciones sensibles al retardo. Una aplicación no notifica a la red su deseo de enviar información simplemente la envía cuando lo ve necesario. Las aplicaciones de FTP File Transfer Protocol - Protocolo de Transferencia de Archivos y HTTP HyperText Transfer Protocol - Protocolo de Transferencia de Hipertexto utilizan este modelo.

1.5.1.2 **Modelo de Servicios Integrados**

IntServ: Integrated Services - Servicios Integrados provee a las aplicaciones servicios “garantizados”, negociando parámetros de red, de extremo a extremo.

La aplicación solicita el nivel de servicio necesario para ella con el fin de operar apropiadamente, y se basa en la QoS para que se reserven los recursos de red necesarios antes de que la aplicación comience a operar. Estas reservaciones se mantienen en pie hasta que la aplicación termina o hasta que el ancho de banda requerido por ésta sobrepase el límite reservado para dicha aplicación.

El modelo IntServ se basa en el protocolo de reservación de recursos (RSVP) para señalizar y reservar la QoS deseada para cada flujo en la red. [2].

1.5.1.3 **Servicios Diferenciados**

En este método se utiliza los enrutadores de bordes para clasificar los tipos de paquetes que circulan por la red, de esta manera se pretende clasificar el tráfico por algunos métodos como por ejemplo utilizando la dirección de red, el protocolo, los puertos o la interfaz de ingreso de cada paquete.

Con este modelo al minimizar la complejidad de la clasificación y el encolado, los ruteadores trabajan a una mayor velocidad. Se minimiza el tráfico de señalización y el almacenamiento ya que cada flujo particular de datos es agrupado en un tipo de clase.

1.5.2 CAMPOS UTILIZADOS PARA IMPLEMENTAR CALIDAD DE SERVICIO

En el protocolo IPv4 se diseño el campo ToS Type of Service- Tipo de servicio para establecer un nivel de prioridad al tráfico de la red, sin embargo no fue mayormente implementado, el protocolo IPv6 incluye en su cabecera los campos Etiquetas de Flujo y Clase de Tráfico para implementar calidad de servicio.

1.5.2.1 **Etiquetas de Flujo**

El campo Etiqueta de Flujo de 20 bits en la cabecera IPv6 puede ser usado por un origen para etiquetar secuencias de paquetes para los cuales solicita un manejo especial por los enrutadores IPv6, tal como la calidad de servicio no estándar o el servicio en "tiempo real".

Este aspecto del IPv6 está, al momento de escribir, todavía experimental y sujeto a cambios. Se exige a los hosts o a los enrutadores que no dan soporte a las funciones del campo Etiqueta de Flujo poner el campo a cero al originar un paquete, pasar el campo inalterado al reenviar un paquete, e ignorar el campo al recibir un paquete. [5]

1.5.2.2 **Clase de Tráfico**

Permite dar un servicio diferenciado entre los distintos paquetes de datos enviados, lo usan los enrutadores para distinguir entre la prioridad de los paquetes IPv6.

Los siguientes requisitos generales se aplican al campo Clase de Tráfico:

- La interface de servicio para el servicio IPv6 dentro de un nodo debe proporcionar un medio para que un protocolo de capa superior proporcione el valor de los bits *Clase de Tráfico* en los paquetes originados por ese protocolo de capa superior. El valor por defecto debe ser cero para todos los 8 bits.

- Los nodos que soportan un uso (experimental o estándar eventual) específico de algunos o todos los bits *Clase de Tráfico* se les permite cambiar el valor de esos bits en los paquetes que ellos originan, reenvían, o reciben, como sea requerido para ese uso específico. Los nodos deben ignorar y dejar sin alterar a cualquiera de los bits del campo *Clase de Tráfico* para los cuales no dan soporte a un uso específico.
- Un protocolo de capa superior no debe asumir que el valor de los bits Clase de Tráfico en un paquete recibido son los mismos que el valor enviado por el origen del paquete.[5]

1.5.3 CALIDAD DE SERVICIO BAJO LINUX

Kernel o núcleo de Linux es el corazón del sistema operativo, su función principal es coordinar el trabajo conjunto entre el software y el hardware del ordenador, además es el encargado de la administración de la memoria para todos los programas y procesos en ejecución, maneja el tiempo de procesador que utilizan los programas y procesos en ejecución, también permite el acceso del usuario a los periféricos/elementos del ordenador, etc.

Las versiones anteriores a la 2.6 del núcleo son la versión de producción y la de desarrollo.

- Versión de producción: es la versión estable hasta el momento, constituye el resultado de varias versiones de desarrollo.
- Versión de desarrollo: es una versión experimental usada para las respectivas correcciones, verificación de nuevas características, comprobaciones, etc.

Las versiones del núcleo antes de la serie 2.6 se numeraban con 3 dígitos luego de ésta se utilizan 4 dígitos.

La forma de numeración es A.B.C

A: Representa la serie/versión principal del núcleo.

B: Indicaba si la versión era de desarrollo ó de producción. Un número impar, significaba que era de desarrollo, uno par, que era de producción.

C: Indicaba el número de revisiones dentro de una versión.

Por ejemplo la versión del núcleo 2.4.1: Núcleo de la serie 2, versión 4, en el que se han corregido errores de programación presentes en la versión 2.4.0.

A partir de la serie 2.6 la numeración consta de 4 dígitos y no existen versiones de producción y desarrollo, son de la forma A.B.C.D donde:

A: Representa la serie/versión principal del núcleo.

B: Indica la revisión principal del núcleo. Números pares e impares no tienen ningún significado hoy en día.

C: Representa revisiones menores del núcleo, como por ejemplo la añadidura de nuevas características y nuevos soporte a drivers.

D: Este dígito cambia cuando se corrigen fallos de programación o fallos de seguridad dentro de una revisión.

A partir de las versiones 2.2.X y 2.4.X Linux provee un gran avance en el área de networking, en esta parte nos enfocaremos a analizar sus ventajas en cuanto a la regulación de ancho de banda para las distintas aplicaciones.

Entre sus múltiples funcionalidades Linux permite regular el ancho de banda de una interfaz de red, realizar un adecuado reparto del mismo de acuerdo a diversos criterios, además permite innovar enrutamiento tan variado y control de tráfico dependiendo del usuario, dirección MAC, dirección IP, tipo de servicio, hora del día y muchas más aplicaciones que en el capítulo III volveremos a tratar.

1.5.3.1 **Linux e Iproute2**

A partir de las versiones 2.2 del kernel (núcleo) de Linux, el subsistema de red ha sido totalmente renovado con respecto a ediciones anteriores, debido a los requerimientos que han surgido continuamente el nuevo código de Linux provee muchas funcionalidades en el área de networking, existen nuevos parches a su implementación de la pila de protocolos TCP/IP, lo que provocó un gran aumento del rendimiento en sus aplicaciones y complejidad en la configuración.

Las facilidades como el ruteo, filtrado y clasificación de tráfico, son más potentes y flexibles que muchos de los productos dedicados que se ofrecen en el mercado actualmente.

La herramienta IProute2 contenida en el kernel de Linux permite realizar las facilidades antes mencionadas, incluye dos herramientas: IP y TC (Control de Tráfico), con esta última podemos configurar, gestionar, clasificar, priorizar, modelar y limitar tanto el tráfico entrante como el saliente.

1.5.3.2 **Funcionamiento de QoS en Linux**

Antes de aplicar QoS es necesario la diferenciación de tráfico para ello se lo divide en clases y colas, en las clases se hace el reparto de ancho de banda mientras que en las colas se envían a la tarjeta de red los paquetes que pertenecen a determinada clase.

Se debe asegurar un reparto equitativo de ancho de banda, para esto se elige un tipo de cola adecuado, para priorizar paquetes que contengan información crítica, se debe ubicarlos en colas cuya prioridad o importancia este por encima de otras.

En la figura 1-15 se indica el proceso de llegada del paquete a la máquina Linux hasta que sale de la misma.

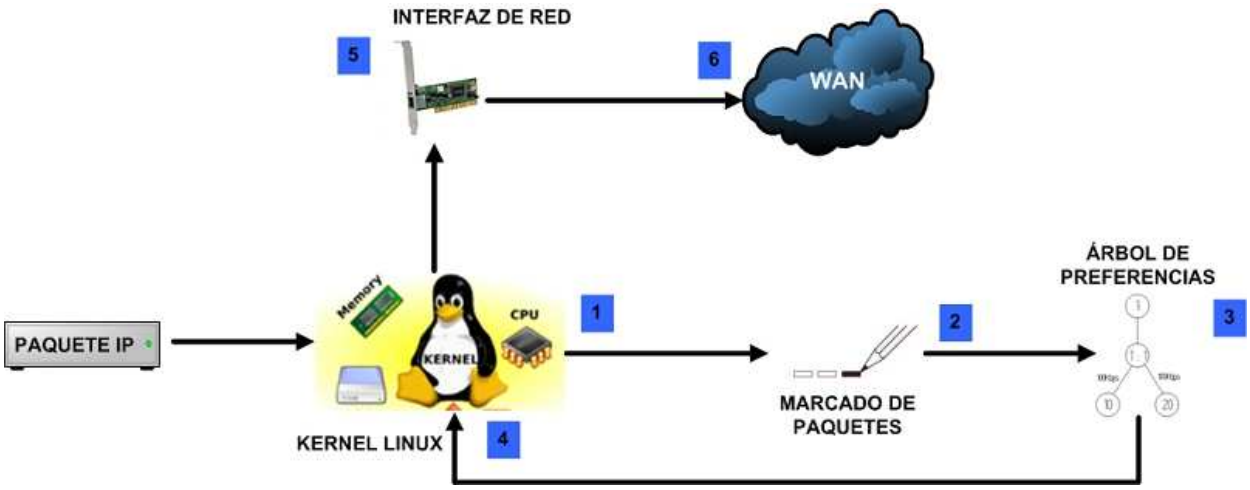

Figura 1- 15 Diagrama de paquetes con disciplinas de colas

Como primer punto el paquete llega al kernel de Linux de la máquina, a continuación Netfilter (Iptables) se encarga de colocar un identificador para clasificar el envío de paquetes, esta marca que se coloca al paquete es establecida por el administrador de la red y puede ser dependiendo del puerto de destino, la cabecera IP, la dirección IP origen, etc.

En el árbol de preferencia se utiliza el criterio de disciplinas de colas para determinar la forma en que se envían los datos.

1.5.4 DISCIPLINAS DE COLAS PARA LA GESTIÓN DE TRÁFICO

En GNU/Linux el control de ancho de banda tiene 2 partes: Un(os) driver(s) o módulos en el kernel y herramientas en espacio de usuario. Estas herramientas se encuentran dentro del paquete iproute2. El control del ancho de banda se logra clasificando los paquetes que tienen que salir por un dispositivo de red en clases y colas. [3]

Ahora bien, cómo se gestionan las colas y qué algoritmo se utiliza para decidir qué paquete desencolamos, la tecnología más utilizada para este control es el uso de Disciplinas de Colas, para su mejor comprensión es necesario tratar algunos conceptos.

1.5.4.1 **Conceptos Disciplina de cola**

Disciplinad de cola (Queueing Discipline, qdisc): Es el algoritmo que se encarga de gestionar el proceso de encolar los paquetes de salida o de entrada en un dispositivo (interfaz de red).

Disciplina de colas sin clases: En esta disciplina de colas no se admite una subdivisión interna que pueda ser configurada por el usuario.

Clases y Disciplinas de colas con clases: Una disciplina de colas con clases puede tener muchas clases. Es un tipo de cola que permite en su interior albergar clases, estas clases pueden ser colas o subclases.

Clasificador y filtro: Las disciplinas de colas con clases necesitan determinar a qué clase envían cada paquete que llega. Esto lo hacen utilizando un clasificador. A su vez la clasificación la llevan a cabo utilizando filtros, los cuales determinan una serie de condiciones que deben cumplir los paquetes. [7]

root qdisc: Esta es la cola que está unida al dispositivo de red (la cola principal).

Scheduling: Cuando en una cola se decide que unos paquetes tienen que salir antes que otros estamos haciendo scheduling.

Shaping: La idea de shaping en encolar y retardar el envío de paquetes de modo de acomodarse a los requerimientos y necesidades de la red.

1.5.4.2 **Disciplinas de colas sin clases**

Los tres procesos para encolar paquetes son pfifo_fast, token bucket filter y stochastic fairness queueing, son procesos sencillos, ya que tienen la capacidad de reordenar, retrasar o descartar los paquetes que van llegando para ser enviados.

1.5.4.2.1 pfifo_fast

Esta disciplina de colas está formada por tres bandas (bandas 0, 1 y 2) las cuales no pueden ser modificadas por el usuario. Dentro de cada banda los paquetes son enviados siguiendo una política FIFO (First In First Out - primero en llegar, primero en ser servido). Existe una prioridad definida para cada banda, la banda 0 tiene la prioridad más alta, mientras que la banda 2 es la de menor prioridad. El campo TOS Type Of Service – Tipo de Servicio de la cabecera IP nos permite determinar los paquetes que van a cada banda.

El proceso consiste en que el kernel de Linux primero da una determinada prioridad a los paquetes en función del valor del campo, y posteriormente existe una mapeo de prioridades, definidas por el usuario, entre la prioridad asignada por el kernel y cada una de las tres bandas.

Sus principales beneficios son la poca carga computacional, al no ser ordenados los paquetes la demora dependerá exclusivamente de la longitud de la cola, entre sus principales inconvenientes son la no diferenciación entre tipos de tráfico, y al existir congestión, la qdisc FIFO beneficia al tráfico UDP sobre el TCP.

1.5.4.2.2 Token Bucket Filter

Este tipo de disciplina de colas permite limitar el ancho de banda de un interfaz determinado, el proceso consiste en suponer que tenemos un buffer al cual llegan los denominados ‘tokens’ a un ritmo constante. Cada paquete IP necesita un token para salir del interfaz de red. En este proceso se presentan tres situaciones determinadas por los ritmos de llegada de tokens y paquetes IP.

- Los paquetes IP llegan al mismo ritmo que los tokens. En este caso, cada paquete IP es asignado automáticamente a cada token y sacado del interfaz.
- Los paquetes IP llegan a un ritmo mayor que el de los tokens, por lo cual deben esperar hasta que esté libre un token para poder salir del interfaz,

si no existen token disponibles durante mucho tiempo los paquetes IP que están en espera comenzarán a ser descartados con lo cual limitamos el ancho de banda.

- Los paquetes IP llegan a un ritmo menor que el de los tokens. En este caso los tokens no utilizados se almacenan en un buffer para su uso posterior. De esta manera si el flujo de paquetes IP aumenta se tendrán tokens disponibles para llevarlos al interfaz de red.

1.5.4.2.3 Stochastic Fairness Queueing

Este tipo de disciplina de colas intenta distribuir el ancho de banda de un determinado interfaz de red de la forma más justa posible. Para ello esta disciplina implementa una política de Round Robin entre todos y cada uno de los flujos de comunicación establecidos en el interfaz, dando a cada uno la oportunidad de enviar sus paquetes por turnos. Un flujo de comunicación será cualquier sesión TCP o flujo UDP, y de esta forma lo que conseguimos es que ninguna comunicación impida al resto poder enviar parte de su información. Lógicamente esta disciplina de colas sólo tendrá sentido en aquellos interfaces que normalmente estén saturados y en los que no queramos que una determinada comunicación evada al resto. [9].

1.5.4.3 **Disciplinas de Colas con Clases**

En estas clases se utilizan filtros para clasificar los paquetes y enviarlos a determinadas clases, estos filtros se asocian a disciplinas de colas, los cuales generan un resultado que permite a la disciplina de colas determinar la clase a la que pertenecerá cada paquete. Las consultas a los filtros son sucesivas para determinar la correcta clasificación del paquete, ya que cada clase tiene asociada una nueva disciplina de colas con o sin clases.

En Linux, cada interfaz de red tiene una disciplina de colas de salida (egress) llamada ‘root’, por defecto esta disciplina es del tipo pfifo_fast.

1.5.4.3.1 Disciplina de colas PRIO

Similar a la disciplina sin clases pfifo_fast, aunque es mucho más versátil y ofrece mayores posibilidades. Aquí se presentan tres clases, cada una de ellas tiene asociada una nueva disciplina de colas con política FIFO. El proceso es el siguiente, al enviar un paquete la cola de clase 1 es la primera en ser atendida, al vaciarse ésta se procesa la cola 2 y posteriormente la cola 3.

El grave problema que presenta es al recibir paquetes continuamente destinados a la clase 1, las clases 2 y 3 no serían atendidas, con una correcta gestión podríamos solucionar este problema tratando de priorizar el tráfico más importante por encima de otro. Para la clasificación de los paquetes, no necesariamente utilizaríamos el campo TOS, existe la posibilidad de definir filtros tan complejos como necesitemos, otra de sus ventajas es la poca carga computacional.

Entre sus desventajas son la ya mencionada desatención a las clases 2 y 3 si existe saturación de paquetes ingresando a la clase 1, además no presenta una solución a la prioridad de los paquetes UDP sobre los TCP. Si se da una mayor prioridad a los paquetes TCP, éstos tratarán de consumir todo el ancho de banda.

1.5.4.3.2 Disciplina de colas Class-Based Queueing (CBQ)

CBQ es una disciplina de cola muy compleja, fue la primera en ser creada y con seguridad es la más utilizada. Maneja una disciplina basada en clases y permite la regulación del ancho de banda para lo cual utiliza un algoritmo basado en la estimación del intervalo de tiempo transcurrido entre dos peticiones consecutivas del hardware para el envío de datos, esto con la finalidad de mantener inactivo un enlace durante un porcentaje de tiempo que asegure que se regula hasta conseguir el ancho de banda deseado.

- Parámetros CBQ para ajustar la regulación del ancho de banda

Avpkt: Tamaño medio del paquete medido en bytes.

Bandwidth: Ancho de banda del dispositivo físico. Se necesita para calcular el tiempo muerto entre petición y petición.

Mpu: Tamaño mínimo de un paquete. Es necesario porque incluso un paquete de cero bytes de datos da lugar a una trama Ethernet de un tamaño mínimo distinto de cero.

Rate: Ancho de banda regulado con el que queremos que funcione nuestra disciplina de colas.

- Parámetros CBQ para definir el comportamiento como disciplina de colas con clases

Al igual que la disciplina PRIO, CBQ permite definir prioridades dentro de las clases que componen su estructura interna. De esta forma, cada vez que el nivel hardware solicita un paquete, CBQ lanza un proceso Round Robin[12] por prioridades. Para la configuración de este proceso CBQ pone a disposición del usuario los siguientes parámetros:

Prio: Establece las distintas prioridades entre las clases que componen la estructura interna de la disciplina de colas.

allot y weight: Ambos parámetros permiten configurar el hecho de que aquellas clases con un mayor ancho de banda puedan enviar mayor cantidad de información cada vez que les toque el turno durante el proceso de Round Robin por prioridades.

[12] Round Robin, Round Robin es uno de los algoritmos de planificación de procesos más simples dentro de un sistema operativo que asigna a cada proceso una porción de tiempo equitativa y ordenada, tratando a todos los procesos con la misma prioridad. [13]

- Parámetros CBQ para definir la posibilidad de prestar y pedir prestado ancho de banda entre las distintas clases

Manteniendo siempre la limitación global en el ancho de banda establecido para la disciplina de colas CBQ, existe la posibilidad de que entre las clases se presten ancho de banda en el caso que sea posible. Los parámetros de que se dispone para ellos son:

isolated/sharing: Una clase configurada con el parámetro 'isolated' no prestará nunca ancho de banda a sus hermanas. El comportamiento contrario viene establecido por el parámetro 'sharing'. Por defecto, si no se indica lo contrario se supondrá que el 'sharing' está activo.

bounded/borrow: Una clase configurada con el parámetro 'bounded' no intentará pedir prestado ancho de banda a ninguna de sus hermanas. El comportamiento contrario viene establecido por el parámetro 'borrow'. Por defecto, si no se indica nada, se supondrá que el 'borrow' está activo.

1.5.4.3.3 Disciplina de colas Hierarchical Token Bucket (HTB)

La disciplina de colas HTB se diseñó con el fin de sustituir a la ya mencionada CBQ, la cual es muy compleja, HTB no forma parte del kernel de Linux, es necesario parchearlo y recompilarlo para poder utilizarla.

El propósito de HTB es controlar el uso de ancho de banda de un enlace dado, realizando un reparto dependiendo de la demanda que tenga cada clase, al utilizar menos AB del asignado se repartirá el resto entre las demás clases que lo pidan. El reparto del ancho de banda sobrante se hace de acuerdo la prioridad de las clases. En base a esto se maneja una jerarquía de clases que pueden resolver el problema de compartir un mismo enlace entre varios tipos de tráfico o incluso entre varias entidades diferentes y dentro de cada una de ellas diferenciando el tráfico y la forma en que se trata.

Una de sus principales desventajas es la necesidad de abundantes recursos computacionales ya que su estructura de compartición del enlace en forma jerárquica provoca un tratamiento sofisticado del tráfico.

1.5.4.4 Utilización de Filtros para la clasificación de paquetes

Para determinar la ubicación de cada paquete en las clases se utiliza filtros asociados a cada disciplina de colas.

1.5.4.4.1 Filtro u32

Es muy utilizado ya que incorpora muchos criterios para lograr la clasificación de los paquetes, este tipo de filtro permite filtrar en función de cualquier conjunto de bits, tanto de la cabecera del paquete IP, como de la cabecera del segmento de datos. Su implementación es muy tediosa y complicada, por lo que normalmente se suelen utilizar formas más directas para estos filtros.

Algunas de estos criterios directos son utilizar la dirección IP de origen/destino del paquete, el protocolo utilizado: tcp, udp, icmp, gre, etc, los puertos de origen y destino utilizados, o el valor del campo ToS de la cabecera IP.

1.5.4.4.2 Filtro route

Este tipo de filtro toma su decisión en función del resultado obtenido al pasar el paquete IP por la tabla de rutas.

1.6 PROTOCOLO IPSEC COMO ESTÁNDAR DE SEGURIDAD A NIVEL DE CAPA RED

El problema de seguridad en capa red, los ataques continuos, virus, engaños fueron razones más que suficientes para implementar seguridad en el envío de información en Internet. El uso de NAT para aliviar la falta de direcciones en IPv4 produjo huecos de seguridad en las comunicaciones ya que al cambiar

puertos y direcciones alteran su contenido y si se usan firmas digitales simplemente las modifican.

Los ataques de reconocimiento y de escaneo de servicios, los ataques Man in the Middle – Hombre en el medio, las inundaciones de información en *broadcasting*, los ataques Smurf y de fragmentación hacen necesario la implementación de este protocolo sin embargo es importante mencionar que en redes no es posible implementar seguridad al 100% una de las razones es por tratar de conseguir dualidad entre ambos protocolos no contemplamos como va a responder IPv4 dentro de IPv6.

Cabe mencionar que una correcta implementación de este protocolo no es suficiente si contamos con una red mal implementada con tiempos de respuesta demasiado largos, inadecuada infraestructura, etc.

El éxito de la seguridad que implementemos dependerá de la fortaleza de los algoritmos criptográficos, de la robustez de la clave, de la seguridad del algoritmo para distribución de claves y de la correcta implementación de cada uno de ellos.

1.6.1 FUNCIONAMIENTO DE IPSEC

IPsec emplea técnicas basadas en criptografía para proporcionar seguridad en la información, se integra en la versión actual de IP (IP versión 4) y se incluye por defecto en IPv6, selecciona el protocolo de seguridad, el algoritmo a usarse y la llave para este propósito.

Protege rutas entre pasarelas[13] (routers, firewall) y servidores o entre servidores.

Utiliza una de las dos cabeceras para proveer seguridad ESP (Encapsulating Security Payload - Encapsulación Segura de Carga Útil) o AH (Authentication

[13] Paraselas, Gateway actúa como intermediario de comunicaciones, inicia y recibe mensajes, sirve servicios IPsec para sí mismo

Header - Cabecera de Autenticación), estas no pueden ser implementadas simultáneamente ya que proveen funciones similares (control de acceso y distribución de llaves para cada flujo de datos) añadida la propiedad de confidencialidad proporcionada por ESP.

Para garantizar estas propiedades combina tecnologías de clave pública (RSA), algoritmos de cifrado (DES, 3DES, IDEA, Blowfish), algoritmos de hash (MD5, SHA-1) y certificados digitales X.509v3.

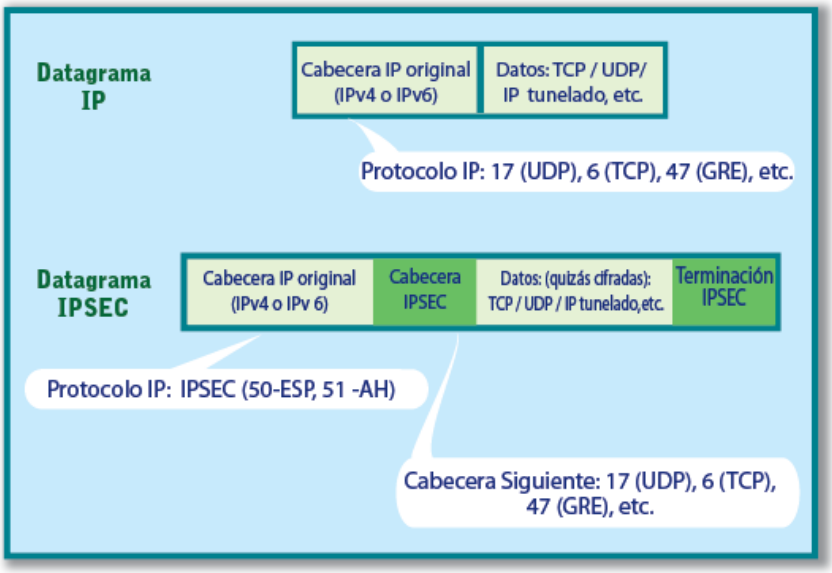

Figura 1- 16 Datagrama IPsec

1.6.2 FUNCIONALIDADES DE IPSEC

Cabecera de Autenticación (AH - Authentication Header): Se implementa en IPv6 con el algoritmo MD5 y generando llaves de 128 bits, provee integridad en el envío de información, esta cabecera no realiza un análisis de tráfico ni provee confidencialidad pero proporciona integridad, autenticidad y no repudio[14] (con algunos protocolos).

[14] No repudio, realiza la función de verificar que la información fue realmente enviada por el emisor y no le permite al emisor luego de enviar dicha información negar que él fue quien la envió

Encapsulación Segura de Carga Útil (ESP - Encapsulating Security Payload): Provee integridad y confidencialidad de la información mediante encriptación de datos y autenticidad si se implementa algoritmos para este propósito.

Intercambio de claves de Internet (IKE - Internet Key Exchange): Es una función que provee generación de llaves. El funcionamiento de estos dos esquemas ESP y AH no afecta el resto de protocolos implementados, son modulares ya que son completamente independientes de toda la pila de protocolos dentro de la red. Sin embargo ESP o AH no se los debe implementar simultáneamente.

1.6.2.1 **Cabecera de Autenticación - Authentication Header**

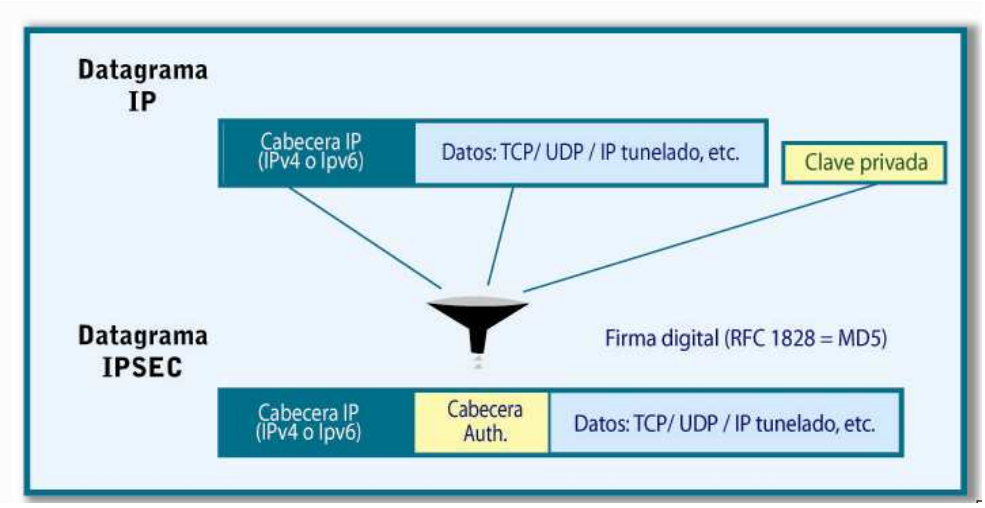

Figura 1- 17 Cabecera de Autenticación

Sobre los datagramas enviados se corre una función (por defecto se implementa el algoritmo MD5) y de acuerdo a una llave secreta generada se autentica.

El procedimiento es el siguiente: el emisor se encarga de autenticarse generando una llave secreta y ejecuta el algoritmo, en recepción se verifica la autenticidad y la integridad de la información recibida.

En este procedimiento se omiten ciertos campos de la cabecera IPv6 o IPv4 que no son requeridos para este proceso y su ausencia no afecta una correcta autenticación además debido a su variación a medida que atraviesa los nodos,

por ejemplo el campo Hop Limit que se va decrementando hasta alcanzar su destino o la Cabecera de Encaminamiento de IPv6 “Next Address” la cual indica la dirección del siguiente nodo a ser alcanzado. En la cabecera IPv4 los campos variables como ToS, TTL, flags, offset y checksum.

AH utiliza el algoritmo de ventana deslizante para prevenir ataques de repetición, la información de usuario, los payloads y las demás cabeceras intervienen en el cálculo de la información de autenticación la cual produce un aumento en los costes de procesamiento y la latencia en comunicaciones, esto se justifica ya que el emisor invierte tiempo en procesar la información de autenticación y al receptor porque compara y autentica la información recibida.

El algoritmo MD5 se implementa y ajusta a protocolos simétricos los cuales no proporcionan no repudio, éste algoritmo se implementa por defecto en la cabecera de autenticación pero podemos usar otros algoritmos por ejemplo asimétricos que proporcionan no repudio cada uno con su llave secreta.

1.6.2.1.1 Formato de Cabecera de Autenticación (Authentication Header)

Figura 1- 18 Formato de Cabecera de Autenticación

- Cabecera Siguiente: Especifica el tipo de protocolo dentro de IP (TCP, UDP, GRE).
- Longitud de Carga Útil: Indica el tamaño de la cabecera de autenticación.

- Reservado: Reservado para uso futuro (hasta entonces todo ceros).
- Índice de parámetro de seguridad - Security Parameter Index(SPI): Permite identificar la Asociación de Seguridad que pertenece este datagrama.
- Número de secuencia: Lleva un control del paquete ya que se va incrementando con cada uno, lo cual ayuda a prevenir los ataques de repetición.
- Autenticación: Contiene el Integrity Check Value - Comprobar la integridad del valor (ICV), el cual es necesario para autenticar el paquete.

1.6.2.1.2 Funcionamiento de Cabecera de Autenticación - Authentication Header

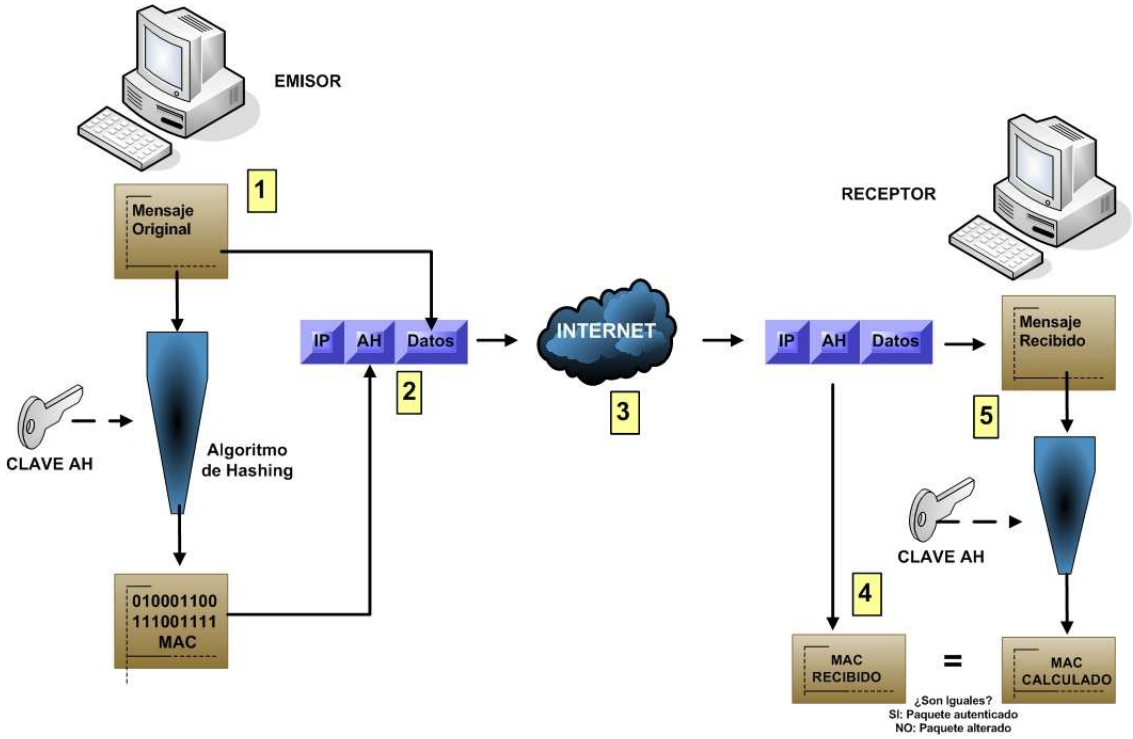

Figura 1- 19 Funcionamiento de Cabecera de Autenticación

El emisor emite un mensaje, el cual es colocado en el campo datos de paquete a transmitirse (1), un extracto de este mensaje original se copia en la cabecera AH del paquete (2), el paquete atraviesa la nube de Internet (3), en el receptor

se identifica el MAC (MAC Message Authentication Code - Código de Autenticación del mensaje) recibido (4). Con los datos del paquete recibido y la clave de AH se procede al cálculo del código de autenticación (MAC) (5) y se verifica si son iguales:

Si MAC recibido = MAC calculado, el paquete se autentica, caso contrario se verifica que el paquete ha sido alterado.

1.6.2.2 Encapsulación Segura de Carga Útil - Encapsulating Security Payload

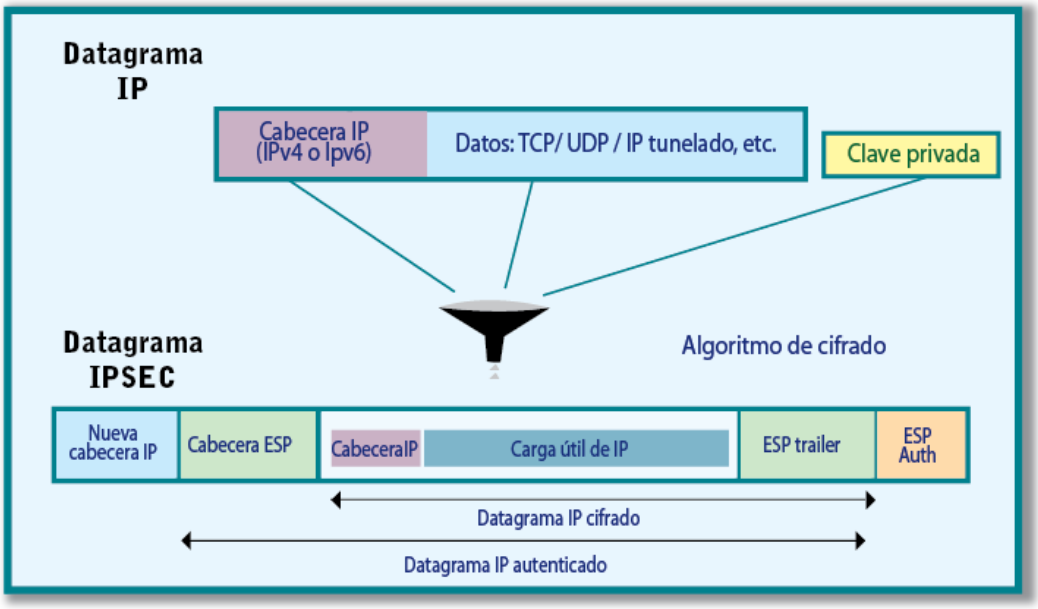

Figura 1- 20 Funcionamiento de Encapsulación Segura de Carga Útil---

Como observamos en la imagen el datagrama IP es procesado y encapsulado dentro de ESP, se puede encapsular todo el datagrama o solo parte de él (información de protocolo de capa superior). Se utiliza una nueva cabecera IP para el envío correcto del datagrama, además se añade una cabecera ESP y un trailer. En recepción se elimina la cabecera nueva de IPv6, a continuación procesa y descifra ESP, elimina la cabecera ESP y así garantiza autenticación, integridad y confidencialidad en la transmisión de información a través del Internet.

ESP al ser implementado aumenta el coste de procesamiento y la latencia en comunicaciones debido al cifrado[15] y descifrado de la información, cuando es implementado en modo transparente (directamente en las máquinas) ahorramos tiempo de procesamiento y ancho de banda ya que solo se encapsula el protocolo de capa superior (TCP o UDP), esto se realiza cuando la confidencialidad del datagrama IP no es importante.

Al implementar ESP en pasarelas se necesita mayor potencia y tiempo de procesamiento ya que el sistema es mucho más complejo que el modo transparente.

ESP ha sido diseñado para soportar DES (Data Encryptión Standard - Estándar de cifrado de datos), pero puede ser implementado con otros algoritmos cabe señalar que el tipo de algoritmo que usemos variará el procesamiento en las redes.

1.6.2.2.1 Formato de datagrama Encapsulación Segura de Carga Útil - Encapsulating Security Payload

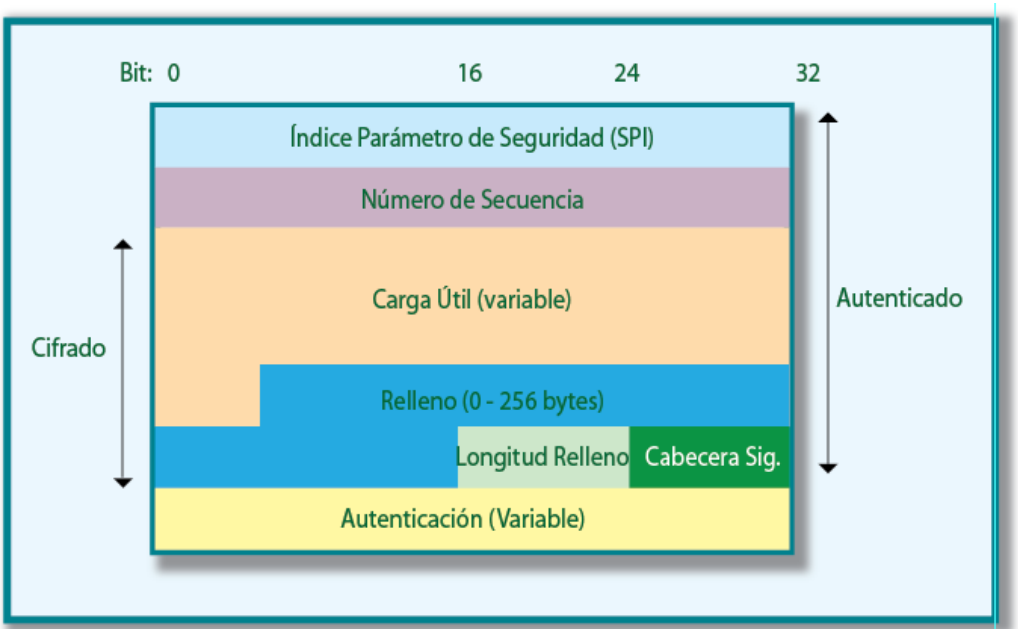

Figura 1- 21 Formato de Encapsulación Segura de Carga Útil

[15] Cifrado, provee confidencialidad

- Índice parámetro de seguridad (SPI): Permite identificar la Asociación de Seguridad que pertenece este datagrama.
- Número de secuencia: Lleva un control del paquete ya que se va incrementando con cada uno, con lo que previene los ataques de repetición.
- Carga útil: Contiene los datos cifrados de protocolo IP.
- Relleno: Es usado por los algoritmos de cifrado para completar los bloques (relleno).
- Longitud del relleno: Indica el tamaño en bytes del Relleno.
- Cabecera siguiente: Especifica el tipo de protocolo dentro de IP (TCP, UDP, GRE).
- Autenticación: Contiene el Integrity Check Value (Comprobar la integridad del valor ICV) calculado sobre todo el datagrama a excepción de campo Autenticación.

Usuarios que necesitan una seguridad más robusta pueden combinar ESP y AH, encapsulando el datagrama IPv6 autenticado (IPv6 con su cabecera de autenticación) dentro de ESP, con esto se lograría tener confidencialidad, integridad autenticación y no repudio (si se utilizan ciertos algoritmos como RSA).

1.6.2.2.2 Funcionamiento del datagrama Encapsulación segura de Carga Útil Encapsulating Security Payload.

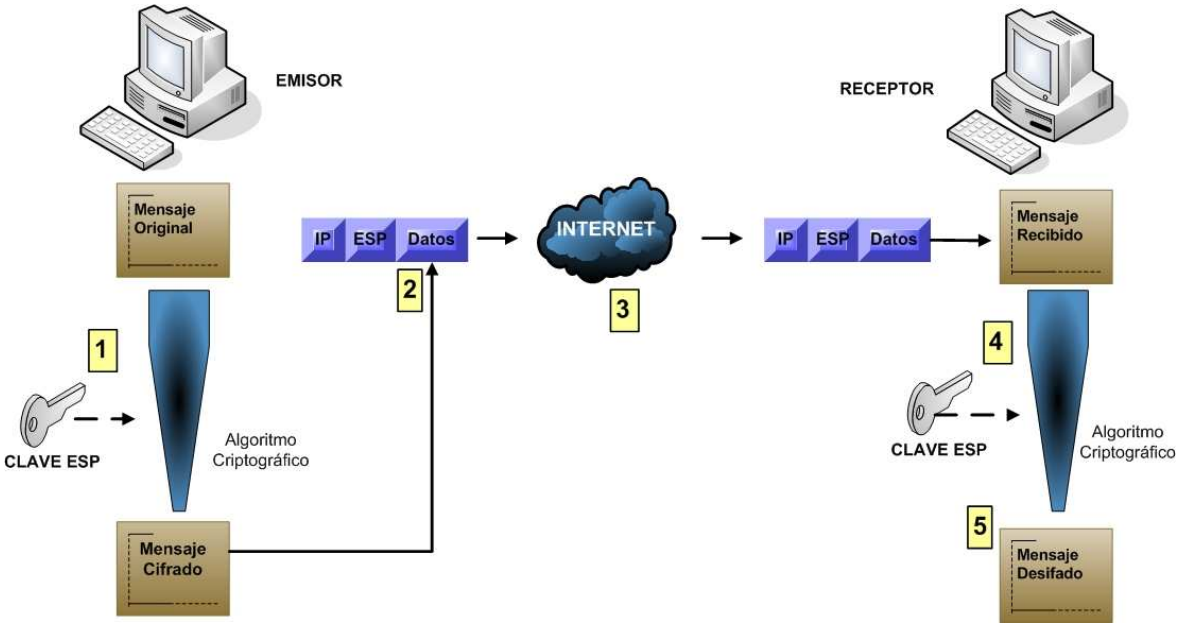

Figura 1- 22 Funcionamiento de ESP Encapsulación Segura de Carga Útil

El emisor envía el mensaje original el cual con la clave ESP y un algoritmo criptográfico es cifrado (1), a continuación el mensaje es colocado en el campo datos del paquete luego de la cabecera ESP (2). El paquete es enviado a través del Internet sin poder ser descifrado ya que nadie conoce la clave para hacerlo (3), en el receptor se aplica el algoritmo de cifrado con la misma clave ESP (4) y finalmente se obtiene el paquete descifrado (5).

La seguridad de este protocolo reside en la robustez del algoritmo de cifrado.

1.6.3 ASOCIACIONES DE SEGURIDAD (AS)

Se implementan Asociaciones de Seguridad mediante el uso de AH o ESP, una AS establece una conexión unidireccional, para tener una vía bidireccional se implementa AS en cada extremo de los equipos a comunicarse.

Asociación de Seguridad es un conjunto de información de seguridad referente a una conexión de red dada (o conjunto de conexiones). Ésta incluye generalmente la clave criptográfica, tiempo de vida de la clave, algoritmo, forma del algoritmo, nivel de sensibilidad (por ejemplo. no clasificada, secreto,

propietario), la clase de servicio de seguridad se proporciona (solamente autenticidad, modo de transporte de cifrado, modo IP de cifrado, o alguna combinación), y posiblemente alguna otra información. [10]

1.6.3.1 **Modalidades de Uso**

Tanto ESP y AH pueden ser implementadas en modo de transporte y en modo túnel.

1.6.3.1.1 Modo de Transporte

No se realiza encapsulamiento del protocolo IP ni lo modifica para proveer seguridad, implica una seguridad primaria de las capas superiores y se lo implementa en sistemas remotos los cuales deben tener implantado IPsec. Representa una AS entre servidores. Al utilizar este modo el contenido del paquete transportado dentro del datagrama AH o ESP son datos de la capa de transporte (por ejemplo, datos TCP o UDP). Como vemos en la figura 1-23 en la sección modo transporte, la cabecera ESP se inserta luego de la cabecera IP original y antes de los datos de capas superiores a los cuales proveerá seguridad.

El modo transporte tiene la ventaja de que asegura la comunicación extremo a extremo, pero requiere que ambos extremos entiendan el protocolo IPSec.

1.6.3.1.2 Modo Tunel

El protocolo IP se encapsula dentro de otro datagrama, se provee una seguridad total y lo modifica para obtenerla. Es implementado en sistemas intermedios (pasarelas) ya que no podrían resolver problemas de fragmentación y construcción del paquete Ipsec. Si una pasarela requiere tener configurado modo transporte es necesario que esta actúe como servidor debido a la complejidad de la implementación. Al utilizar este modo el contenido dentro del datagrama AH o ESP es un datagrama IP completo, por

ello el incremento de una nueva cabecera IP para el enrutamiento a través de la red. En la figura 1-23 en la sección de modo túnel, se distingue como el datagrama IP completo es precedido por la cabecera ESP y la adicción de una nueva cabecera IP para el enrutamiento.

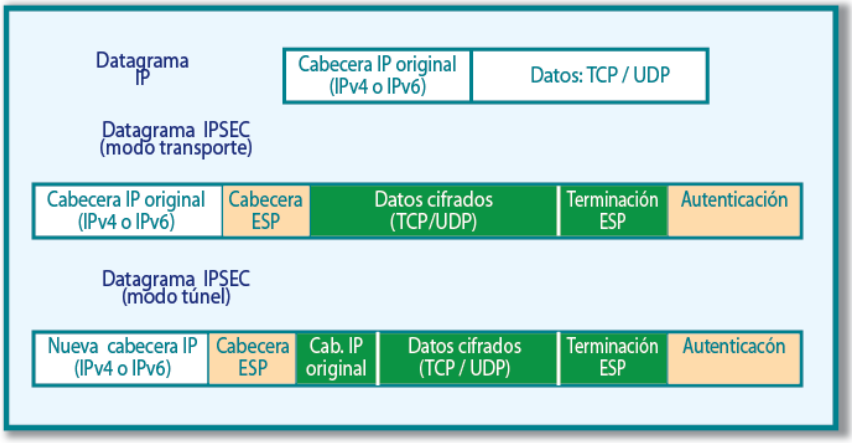

Figura 1- 23 Modo Transporte vs. Modo Túnel

1.6.3.2 **Parámetros SA**

Una Asociación de Seguridad define un conjunto de parámetros:

- Índice de parámetros de seguridad (SPI): Su significado es local y permite apuntar a la SPD (Security Police Database - Política de Seguridad de Base de Datos).
- Direcciones IP destino: Permite definir solo direcciones unicast.
- Identificación protocolo de seguridad: Define la utilización de ESP o AH.
- Secuencia de número: Permite generar un número de secuencia transmitido en las cabeceras ESP y AH, su valor es de 32 bits.
- Secuencia ante un overflow: Es un indicador de acción ante un llenado de número de secuencia.
- Ventana limitación: Parámetro para limitar la aceptación de datagramas válidos.
- Información AH: Contiene información de autenticación, algoritmo usado, la llave y el tiempo de vida.
- Información ESP: Contiene información de confidencialidad, algoritmo

usado, la llave y el tiempo de vida.

- Modo Ipsec: Definen el modo de transporte del datagrama Ipsec.(Transporte o Túnel)
- Tiempo de vida SA: Tiempo en el que una SA se considera válida. Luego de este tiempo se reemplaza por una nueva SA.
- Ruta MTU: Define la unidad máxima de transferencia de la información transmitida sin fragmentación.

1.6.3.3 **Funcionalidad de SA**

La seguridad que brinde una SA va a depender de los parámetros escogidos por el administrador es decir si se utiliza AH o ESP, que algoritmos de seguridad se van a implementar, el modo escogido para el transporte de la información (transporte o túnel).

El protocolo de encriptación influye notablemente en el rendimiento que ESP proporcione al datagrama.

Cada SA define algoritmos criptográficos, estos dependen de las políticas de seguridad que los usuarios determinen para la información a enviar, por defecto se implementan algoritmos simétricos como DES para cifrar la información y para autenticación por lo menos Ipsec debe soportar MD5 o SHA-1.

1.6.4 ADMINISTRACIÓN DE CLAVES

Los algoritmos usados para la administración de claves en la SA son independientes del protocolo escogido para proveer seguridad (AH o ESP), debido a que la información sobre el manejo de llaves la proporciona el protocolo de capas superiores, esta técnica es conocida como administración de claves "out-of-band", sin embargo es obvio que afectan a cada uno de ellos.

Existen dos técnicas que nos proporcionan el manejo de llaves a través del Internet, una muy sencilla utilizada para ambientes pequeños y estáticos configurada manualmente y una dinámica que requiere mucha más

complejidad en su uso.

1.6.4.1 **Distribución Manual de Claves**

Se emplea en sistemas estáticos y de pequeña escala, en donde el administrador configura cada sistema para que tenga claves y SA's de los otros esquemas y suyas, esto es vital para conectar usuarios mutuamente desconfiados. Su funcionalidad principal es la de brindar seguridad a ciertas comunicaciones previamente determinadas, además se recomienda el uso de algoritmos simétricos para este esquema.

1.6.4.2 **Distribución Automática de Claves**

IKE (Internet Key Exchange – Intercambio de claves de Internet) es el protocolo estándar seleccionado para el manejo de claves a través del Internet permite autenticación de extremo a extremo, generación automática de claves y negociación de servicios de seguridad para ello es necesario que ambos nodos estén de acuerdo tanto en los algoritmos criptográficos como en los parámetros de control. Utiliza la infraestructura de clave pública.

El proceso de autenticación provisto por IKE comprende una autenticación basada en una clave pre-compartida, o una autenticación basada en encriptación asimétrica (empleando certificados digitales).

Este mecanismo de distribución de claves es mucho más flexible sin embargo los requerimientos de software y la complejidad de configuración es mayor.

Es un protocolo híbrido compuesto de dos protocolos complementarios ISAKMP (Internet Security Association and Key Management Protocol - protocolo para el establecimiento de asociaciones de seguridad) y Oakley (protocolo para la determinación de claves).

ISAKMP define de forma genérica el protocolo de comunicación y la sintaxis de los mensajes que se utilizan en IKE, mientras que Oakley especifica la lógica de cómo se realiza de forma segura el intercambio de una clave entre dos

partes que no se conocen previamente. [4].

Se puede utilizar otros protocolos que brinden una adecuada distribución de llaves, éstos deben adaptarse a los requerimientos de Ipsec, un correcto manejo de mismos garantiza una correcta seguridad.

1.6.4.3 **Infraestructura de Clave Pública (PKI)**

La solución que ofrece los mecanismos y elementos necesarios de gestión de información criptográfica para el establecimiento de comunicaciones seguras es lo que se llama PKI[16] (Public Key Infrastructure - Infraestructura de Clave Pública).

Una vez que los certificados han sido emitidos, los administradores pueden establecer canales seguros entre las redes. Es en este momento cuando el software de IKE obtendrá la información criptográfica necesaria para la negociación, ya sea la clave privada a través de la tarjeta inteligente o su propio certificado o el del otro extremo mediante el acceso al directorio de certificados.[4]

1.7 APLICACIONES DE IPSEC - REDES PRIVADAS VIRTUALES

IPsec se puede configurar para cifrar la información entre dos servidores (modo transporte), o para construir túneles virtuales a través del Internet, esto se conoce como redes privadas virtuales, en las cuales se pretende que diferentes redes se comporten e intercambien información como si se tratara de una única

[16] PKI, Bajo el nombre de PKI (Infraestructura de Clave Pública) se engloban todos los elementos y procedimientos administrativos que permiten emitir, revocar y, eventualmente, renovar los certificados digitales para una comunidad de usuarios [4].

red, es decir que las máquinas de cada red tengan acceso a las carpetas compartidas como si estuvieran conectadas a un único ruteador.

Una VPN (Red Privada Virtual), provee conectividad segura sobre una red pública, nos proporciona menor coste que las redes privadas dedicadas ya que su infraestructura es compartida.

Su utilización es importante en empresas separadas geográficamente las cuales necesitan compartir datos de forma segura, en extranets y en usuarios móviles que necesitan acceder a la red de la empresa desde cualquier lugar y de forma segura. Los servicios que proporciona son múltiples como la implementación de Voz IP, servicios críticos como son la videoconferencia entre PC´s, permite la compartición de recursos del servidor así como también los archivos y equipos (impresoras) de la red.

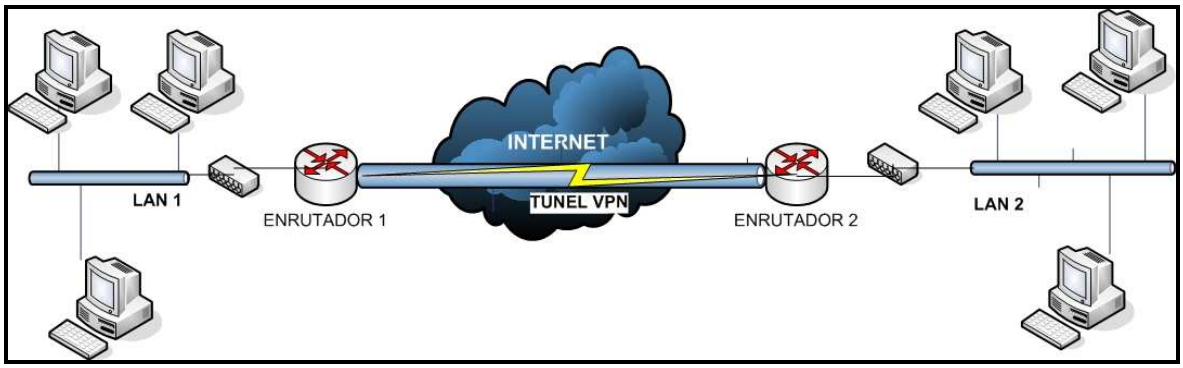

Figura 1- 24 Red Privada Virtual

1.7.1 TECNOLOGÍAS DE LAS VPN

- PPTP (Point-to-Point Tunneling Protocol - Protocolo de Túnel Punto a Punto), se encapsula las tramas PPP en datagrams IP, la conexión de control se realiza sobre TCP en el puerto 1723 es una tecnología creada por Microsoft.
- L2TP (Layer 2 Tunneling Protocol - Protocolo Túnel capa 2), permite encapsular tramas PPP sobre cualquier medio, no necesariamente redes IP. Si se utilizara IP la conexión es sobre UDP en el puerto 1701.Tiene problemas en el manejo de seguridad e incompatibilidades.

- IPSec, integra confidencialidad, integridad y autentificación, con la llegada de ipv6 es una de las mejores opciones para la implementación de las VPN. Funciona directamente sobre IP, se realiza una validación de usuarios utilizando certificados X.509 (SSL), claves secretas compartidas por ambos extremos o claves RSA. Para la negociación de la conexión utiliza el protocolo ISAKMP.

Los paquetes IP a transmitirse en una VPN se cifran para garantizar la confidencialidad, se firman para garantizar la autenticidad e integridad, de esta manera el paquete resultante se encapsula en un nuevo paquete IP y se envía a través de la red insegura al otro extremo de la VPN:

1.7.2 REQUERIMIENTOS BÁSICOS DE UNA VPN

- Identificación de usuario: Verificación y autenticación de los usuarios que acceden a la VPN.
- Administración de direcciones: La VPN debe establecer una dirección del cliente en la red privada y debe cerciorarse que las direcciones privadas se conserven así.
- Codificación de datos: Los datos se envían encriptados por la red pública.
- Administración de claves: La VPN debe generar y renovar las claves de codificación para el cliente y el servidor.
- Soporte a protocolos múltiples: La VPN debe ser capaz de manejar los protocolos comunes que se utilizan en la red pública.[4]

Una red privada virtual nos ofrece múltiples ventajas como el envío de información íntegra, garantizando la confidencialidad de la misma, provee mecanismos de control de acceso basado en políticas de la organización así como herramientas de diagnóstico remoto.

Al implementar algoritmos de compresión permite mejorar el tráfico del cliente y disminuye el costo de mantenimiento a los PC´s remotos, evitando también el alto costo de las actualizaciones.

Para el establecimiento de una VPN los usuarios remotos dispondrán de un software instalado en su computador el cual permite su acceso y el establecimiento de una conexión segura con la red local de la compañía.

Los actuales sistemas operativos incluyen un cliente IPsec para el envío de información segura a través del Internet, si se necesita otro que no contempla esta funcionalidad existen aplicaciones de cliente IPSec tanto comerciales como de libre distribución.

CAPÍTULO 2. ANÁLISIS DE SOFTWARE PARA LA IMPLEMENTACIÓN DE LOS PROTOCOLOS

2.1 INTRODUCCIÓN

El éxito que Internet ha tenido en los últimos años ha impuesto una serie de presiones ante las limitaciones que posee la arquitectura TCP/IP[17]. Entre estas se encuentran, el limitado espacio de direccionamiento de la versión 4 del protocolo IP[18], pobre calidad de servicio, falta de soporte a la movilidad, transmisión de datos poco confiables y sin políticas de seguridad definida.

En la actualidad las plataformas basadas en Linux han tomado gran notoriedad gracias a los desarrollos tan acertados por parte de la empresa privada, centros de estudio y la comunidad de colaboradores, todos motivados por la libertad del código y de la fortaleza del mismo para crear sistemas robustos, confiables y con proyección para solucionar estos problemas que cada día son más complicados de manejar.

Por lo que en el presente capítulo se ha realizado un estudio detallado y minucioso de las distribuciones más importantes y posicionadas en el mundo con el objetivo de seleccionar la más adecuada como base para la implementación de software de enrutamiento que permita establecer un ambiente propicio, estable y confiable para levantar servicios de enrutamiento, con el objetivo de conformar una solución de enrutamiento basada en software, de última tecnología capaz de estar listo y acorde con las nuevas tecnologías para redes y sin la necesidad de realizar pagos por licencias.

---

[17] TCP/IP es el protocolo común utilizado por todos los computadores conectados a una red de datos, de manera que éstos puedan comunicarse entre sí.

[18] IP (Internet Protocol) El Protocolo de Internet es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de datos.

La seguridad también es un aspecto tratado en este capítulo, ya que actualmente con el crecimiento del Internet y su naturaleza pública lo hace un medio inseguro de trasmisión de datos por lo que también se realiza un análisis de las opciones más importantes para la implementación de protocolos de seguridad IP.

La selección se realizará con la ayuda de la Norma de Selección IEEE 830, para la elegir del software más adecuado para la implementación del proyecto.

2.2 ANÁLISIS DE LAS DISTRIBUCIONES DE LINUX EXISTENTES

2.2.1 FEDORA

Figura 2- 1 Símbolo de Fedora

2.2.1.1 Características Generales

Fedora Core es un sistema operativo de propósito general que no se concentra en un mercado específico. Es apto para ambientes domésticos, ambientes de programación y como servidor empresarial. En cada caso se requiere un proceso de personalización, lo que es una desventaja generada por tratar de ser bueno para todos los usuarios.

Fedora intenta ser una distribución innovadora y fresca que incluye muchos paquetes de tecnologías de punta y software de prueba, fue la primera distribución que implementó software de máquina virtual de código abierto.

Esta inclinación por ser innovadores proporciona muchas ventajas, pero también genera una gran desventaja, que es la potencial inestabilidad, por ser

software de prueba. Por lo tanto, si se desea probar herramientas nuevas y novedosas esta distribución es una buena opción, pero es importante tener en cuenta las posibles consecuencias.

Fedora es una distribución muy versátil y también útil para desarrolladores, pues, contiene en sus discos de instalación las herramientas de programación más destacadas, lo que las hace bastante fáciles de ser instaladas al sistema operativo. Los desarrolladores de Java cuentan con la inclusión de Eclipse IDE[19] y muchas bibliotecas Java para el desarrollo web, también están disponibles sin costo algunos paquetes como GCC[20] 4.x (GNU Compiler Collection, Colección de Compiladores GNU), Mono[21] y herramientas de desarrollo de aplicaciones web.

Fedora ha sido siempre una distribución que incluye GNOME[22] y KDE[23](K Desktop Environment, Entorno de Escritorio K), aunque en los discos incluyen escritorios muy populares como XFCE[24](XForms Common Enviroment, Ambiente Común XForms), Fluxbox[25], etc., que están disponibles también en

---

[19] Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE). [11]

[20] GCC es un conjunto de compiladores creados por el proyecto GNU, estos compiladores se consideran estándar para los sistemas operativos derivados de UNIX. [12]

[21] Mono es un proyecto para crear un grupo de herramientas libres, basadas en GNU/Linux y compatibles con .NET.

[22] GNOME es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix/GNU/Linux, compuestos enteramente de software libre.

[23] KDE es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

[24] Xfce es un entorno de escritorio ligero para sistemas tipo Unix como Linux, BSD, Solaris y derivados.

[25] Fluxbox es un gestor de ventanas para el Sistema X Window. Su objetivo es ser ligero y altamente personalizable, con sólo un soporte mínimo para íconos, gráficos y capacidades básicas de estilo para la interfaz.

los repositorios de Fedora. El usuario puede escoger su escritorio favorito durante la instalación.

2.2.1.2 **Características Técnicas**

Ítem	*Características*	*Detalles*	*Valoración Comparativa* (*)
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC.	6
2	**Requisitos de hardware mínimos**	Modo texto: *Procesador: 200 MHz Pentium,* *Memoria: 64 MB,* *Disco Duro: 620 MB.* Modo gráfico: *Procesador: 400 MHz Pentium,* *Memoria: 192 MB,* *Disco Duro: 620 MB.*	10
3	**Licencia**	El Software básico es GNU GPL pero incluye paquetes con software propietario.	5
INSTALACIÓN			
4	**Instalador Global**	El instalador es muy desarrollado, ofrece funciones en modo texto y modo gráfico tanto para usuarios principiantes como para expertos. Contiene opciones para la personalización de la instalación pero son limitadas. El defecto más notable en la lenta velocidad de la instalación.	7
5	**Selección de Paquetes**	Los paquetes pueden ser seleccionados fácilmente e incluyen todas las dependencias, descargadas de los repositorios oficiales y los discos de instalación.	10
6	**Grupo de paquetes predefinidos**	Todos los grupos de paquetes predefinidos incluyen paquetes instalados por defecto y opcionales.	9
7	**Instalación en Modo Experto**	La mayoría de las pantallas incluyen opciones "avanzadas" que permiten personalizar las	7

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

		configuraciones no estándares.	
8	**Instalación Gráfica**	La instalación gráfica utiliza Anaconda, que es una herramienta sencilla e intuitiva que permite realizar configuraciones personalizadas.	10
9	**Velocidad de Instalación**	El tiempo de instalación es extenso, aunque eso depende de la cantidad de paquetes seleccionados, el sistema básico lleva aproximadamente 2 horas.	4
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	Permiten la configuración en modo gráfico basado en GNOME. La mayor parte de configuración del sistema operativo se puede realizar sin necesidad de abrir la ventana de terminal.	5
11	**Manejo del Sistema Basado en Consola**	Fedora contiene algunas herramientas de consola que permiten la gestión del equipo de manera integral, incluyen la configuración de la tarjeta de red, audio, video, servicios, etc.	10
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	El número de paquetes incluidos es cerca de 12000 que es mejor que los contenidos en la distribución de SUSE Linux, pero no es tan extensa como las distribuciones de Mandriva o Debian.	8
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	Posee un gestor de paquetes por defecto llamado yum[26] que facilita la descarga de paquetes con sus respectivas dependencias.	10
14	**Herramientas Gráficas de Manejo de Paquetes**	Fedora Core cuenta con herramientas gráficas basadas en yum como son: Pirut para realizar el manejo de paquetes y Pup para realizar las actualizaciones. Desde Fedora Core 6 se proporciona una herramienta de notificación de actualizaciones llamado Puplet y Synaptic.	10
EFICIENCIA			
15	**Velocidad del**	El tiempo de arranque de Fedora cuando los	5

[26] YUM es una herramienta de software libre de gestión de paquetes para sistemas Linux basados en RPM.

	Sistema de Arranque	scripts están escritos correctamente tiene una velocidad promedio de 3 minutos para el sistema por defecto.	
16	**Velocidad de Respuesta del Sistema**	La capacidad de respuesta, considerando la velocidad es aceptable, y no existen configuraciones especiales de optimización para uso de escritorio o de servidor.	5
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	Fedora Core ofrece características de seguridad adicionales como: exec-Shield[27], chequeo de compilación en tiempo de buffer de memoria, endurecimiento de datos y acceso restringido núcleo de memoria y recursos de red.	10
18	**Estabilidad y Madurez**	Fedora Core es una distribución de prueba de software de Red Hat comparable con distribuciones como Ubuntu o SUSE linux.	5
19	**Documentación**	Los procedimientos adicionales de localización de documentación y manuales son fácilmente accesibles especialmente en la página oficial del proyecto. (Documentos, respuestas a preguntas frecuentes).	10

Tabla 2- 1 Características técnicas de GNU/Linux Fedora

2.2.2 UBUNTU

Figura 2- 2 Símbolo de Ubuntu

2.2.2.1 Características Generales

Ubuntu es una distribución libre GNU/Linux patrocinado por Mark Shuttleworth y Canónica Ltda. Se centra en la facilidad de uso, soporte de hardware y

[27] Exec Shield es un proyecto realizado por Red Hat, con el objetivo de reducir el riesgo de gusanos u otros ataques automatizados en sistemas Linux.

funcionalidad, es actualmente una de las distribuciones más populares de GNU/Linux con amplia documentación y variadas comunidades en línea.

La comunidad se fundamenta en las ideas consagradas en la filosofía Ubuntu: que se basa en que el software debe estar disponible gratuitamente, las herramientas de software deben ser utilizables por la gente en su lengua local, debe ser posible su utilización a pesar de cualquier discapacidad, y que la gente debe tener la libertad de personalizar y modificar de la forma que consideren más adecuada.

Ubuntu es rápido y fácil de instalar gracias al Disco Vivo (LiveCD), cuenta con emisiones regulares y previsibles, cada 6 meses con Soporte de Largo Tiempo (LTS) de 1,5 años.

Programas especializados pueden ser agregados fácilmente usando los repositorios y cuenta con un buen apoyo de la comunidad por medio de listas de correo, canales IRC[28] y foros web.

La interfaz de usuario de Ubuntu (escritorio GNOME) es muy coherente y estética. Adicionales a las aplicaciones estándar de GNOME, se incluyen por defecto algunos paquetes externos de código abierto, como OpenOffice[29], Firefox[30] y GIMP[31] (GNU Image Manipulation Program, Programa de Manipulación de Imágenes GNU). El sistema detecta automáticamente los dispositivos móviles como, memorias USBs, cámaras digitales y tarjetas de

---

[28] IRC - Internet Relay Chat, protocolo de comunicación en tiempo real basado en texto.

[29] OpenOffice.org es una suite ofimática de software libre y código abierto de distribución gratuita que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos.

[30] Firefox es un navegador de Internet libre y de código abierto descendiente de Mozilla Application Suite, desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos.

[31] GIMP es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías.

memoria. Su gran capacidad de detección de hardware son especialmente visibles en los equipos portátiles, donde tecnologías como el WiFi, software de suspensión y ahorro de energía que solían causar problemas en sistemas GNU/Linux, ahora son muy fácilmente utilizables.

Ubuntu no es sólo de GNOME, otros entornos de escritorio están disponibles también: Ubuntu con KDE (Kubuntu), XFCE (Xubuntu), Fluxbox (Fluxbuntu), etc.

2.2.2.2 **Características Técnicas**

Ítem	*Características*	*Detalles*	*Valoración Comparativa*(*)
1	**Arquitecturas Soportadas**	I386, IA64, AMD64.	3
2	**Requisitos de Hardware Mínimos**	Modo texto: *Procesador: 400 MHz Pentium,* *Memoria: 192MB,* *Disco Duro: 450MB.* Modo gráfico: *Procesador: 800 MHz Pentium,* *Memoria: 256MB,* *Disco Duro: 2GB.*	4
3	**Licencia**	El Software básico es libre GNU GPL pero incluye paquetes con software propietario.	5
INSTALACIÓN			
4	**Instalador Global**	El programa de instalación es rápido y realiza muy pocas preguntas. Únicamente se añaden unas pocas pantallas para el modo experto, y elimina algunas en modo principiante, por lo que limita la posibilidad de personalización.	3
5	**Selección de Paquetes**	La selección de paquetes es muy deficiente, pues no está disponible. Sin embargo, se puede instalar paquetes adicionales usando apt-get.	1

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

6	**Grupo de paquetes predefinidos**	Únicamente se puede seleccionar entre opciones de escritorio o servidor. No se puede realizar la selección por grupos de paquetes.	2
7	**Instalación en Modo Experto**	Posee dos opciones de instalación: Modo Experto (núcleo 2.4) y Principiante (núcleo 2.6).	3
8	**Instalación Gráfica**	Los cuadros de diálogo del instalador son bastante simples de entender y es muy amigable para usuarios novatos. Las instalaciones personalizadas no cuentan con facilidades.	3
9	**Velocidad de Instalación**	El Live-CD de instalación es bastante rápido, realiza pocas consultas y luego copia todo el Live-CD en el disco, configura el hardware e inicia el menú de arranque, proceso que lleva alrededor de 30 minutos.	9
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	Ubuntu no proporciona un panel de control específico para su distribución. Aún así cuenta una gran cantidad de herramientas específicas agregadas por defecto como: notificador de actualizaciones, gestor de instalación y actualizaciones, gestor de red para wifi y gestores de búsqueda.	10
11	**Manejo del Sistema Basado en Consola**	Este paquete de herramientas de configuración llamado debconf, es muy útil, proviene del proyecto Debian y está disponible en las herramientas de configuración estándar de Debian.	10
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	Existen repositorios oficiales, con más de 10000 paquetes específicos para Ubuntu, construido y soportado por su equipo de desarrollo.	8
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	Smart es un gestor de paquetes utilizado en Ubuntu es el único gestor que se considera superior a APT por su rapidez. Permite encontrar los paquetes certificados y no certificados desde los repositorios de la distribución en internet y resolver automáticamente las dependencias.	10
14	**Herramientas Gráficas de Manejo de Paquetes**	Sináptica es una interfaz gráfica para Smart y es una herramienta de actualización muy útil si los usuarios prefieren hacer clic rápidamente.	10

EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	Considerando la selección de los servicios y la configuración por defecto el arranque de Ubuntu lleva en promedio un tiempo de 1 minuto.	8
16	**Velocidad de Respuesta del Sistema**	Cuenta con configuraciones especiales de optimización habilitadas manualmente para utilización especialmente como estaciones de trabajo.	5
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	Todos los paquetes de seguridad importante incluyendo los paquetes de protección de memoria y núcleo que se actualizan a diario, sin embargo, no existen cortafuegos y casi ninguna herramienta de seguridad acceso a recursos de red en la instalación por defecto.	8
18	**Estabilidad y Madurez**	Ubuntu está basado en Debian, que es una de las distribuciones más estables y maduras disponibles actualmente. Sin embargo, Ubuntu incluye software adicional y las situaciones de inestabilidad pueden ocurrir.	5
19	**Documentación**	Permite obtener fácilmente la documentación necesaria, desde varias fuentes, como: la página oficial del proyecto, las comunidades y los manuales del sistema.	7

Tabla 2- 2 Características técnicas de GNU/Linux Ubuntu

2.2.3 DEBIAN

Figura 2- 3 Símbolo de Debian

2.2.3.1 Características Generales

Debian es una distribución estable, madura y popular. Ofrece una excelente herramienta de gestión de paquetes – APT Advanced Packaging Tool, Herramienta Avanzada de Empaquetado y el mayor repositorio de software libre de todos los sistemas operativos. Muy configurable y después de un trabajo de personalización, es un sistema operativo muy amigable. Pero necesita obligatoriamente de algunos conocimientos.

Del contrato social de Debian se describen los principales aspectos del sistema. Se dice que Debian es, y siempre será un sistema operativo libre. Sólo el software libre se encuentra en el repositorio principal. Programas que requieren componentes restringidos para su correcto funcionamiento están en el repositorio contrib, y el software no libre se encuentra en repositorios de software no-libre.

Debian usa su propio formato de paquetes .deb. Los paquetes para módulos se mantienen por medio de la herramienta dpkg. La resolución de dependencias y otras instalaciones de alto nivel son proporcionadas por APT que es la interfaz por defecto. También hay disponibles módulos de actualización que permite a las dependencias obsoletas eliminarse fácilmente y no permiten que queden programas huérfanos de bibliotecas.

Debian es una distribución GNU/Linux muy popular, si se incluye también a todos los derivados de Debian, es claramente la más popular de las distribuciones. Debido a la gran popularidad y la comunidad multilingüe, es muy fácil obtener ayuda para Debian y sistemas basados en Debian. Hay muchos foros relacionados con Debian, Usnet, IRC y listas de correo.

2.2.3.2 Características Técnicas

Ítem	*Característica*	*Detalles*	*Valoración Comparativa(*)*
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC, ALPHA, MIPS, MIPSEL, SOURCE, MULTI-ARCH, ARM.	10
2	**Requisitos de Hardware Mínimos**	Modo texto: Procesador: 400 MHz Pentium, Memoria: 32MB, Disco Duro: 450MB. Modo gráfico: Procesador: 800 MHz Pentium, Memoria: 64MB, Disco Duro: 450MB.	4
3	**Licencia**	La licencia es completamente software libre GNU GPL sin embargo, Debian contiene repositorios no libres que no incluyen por defecto, sino que hay que descargarlas manualmente.	10
INSTALACIÓN			
4	**Instalador Global**	Para los usuarios expertos existen más opciones disponibles para que todos puedan elegir su forma de instalar. En general el instalador de Debian es muy funcional, maduro y libre de errores. El instalador gráfico aun está en versión beta, pero hace este proceso aún más amigable para los usuarios no técnicos y permite las personalizaciones en todos los pasos de la instalación.	10
5	**Selección de Paquetes**	Aptitude[32] está disponible durante la instalación, todos los paquetes están agrupados, bien descritos y pueden incluir o eliminar paquetes adicionales, los paquetes incluyen las dependencias.	10
6	**Grupo de paquetes predefinidos**	Existe la posibilidad de escoger los grupos de paquetes, como: entorno de escritorio, sistema de base de datos, el entorno de programación y entorno Web.	4
7	**Instalación en**	Existe el modo Experto y Principiante, y las	7

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

[32] Aptitude es un gestor de paquetes que proporciona la funcionalidad de instalación de paquetes y resolución de dependencias.

	Modo Experto	personalizaciones están disponibles en varias ocasiones durante la instalación.	
8	**Instalación Gráfica**	El instalador gráfico (todavía en fase beta) es una copia funcional de la consola del instalador en GTK+[33] que es muy amigable con opciones de personalizar la instalación.	7
9	**Velocidad de Instalación**	El proceso de instalación del sistema base es bastante rápido, lleva de 15 a 30 minutos.	9
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	No hay herramientas especiales.	1
11	**Manejo del Sistema Basado en Consola**	Contiene un muy buen paquete de herramienta de configuración (debconf), permite la configuración de todos los parámetros del sistema. El sistema de paquetes puede ser modificado mediante dpkg.	10
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	Cerca de 18 mil paquetes en el repositorio principal. Debian es un líder absoluto en ese sentido.	10
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	APT - herramienta de gestión de paquetes de Debian es un líder entre todas las herramientas para la gestión de paquetes de GNU/Linux. La instalación de software en Debian es fácil y sin esfuerzo ya que resuelve las dependencias automáticamente.	10
14	**Herramientas Gráficas de Manejo de Paquetes**	Sináptica, una aplicación gráfica de instalación de software y herramienta de actualización, es una aplicación de APT. Muy útil si alguien le gusta nada más que hacer clic.	10
EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	El sistema de arranque es muy rápido. El uso de update-rc.d, herramienta para eliminar los servicios innecesarios, pueden ayudar a crear el Sistema Operativo más rápido aun con un tiempo	9

[33] GTK es una biblioteca del equipo GTK+, la cual contiene los objetos y funciones para crear la interfaz gráfica de usuario. Maneja ventanas, botones, menús, etiquetas, deslizadores, pestañas, etc.

		de 40 a 50 segundos en una instalación por defecto.	
16	**Velocidad de Respuesta del Sistema**	La velocidad de respuesta se encuentra en un nivel medio. Especialmente en los programas que no han cumplido con las optimizaciones por defecto, sin embargo existe la posibilidad de utilizar scripts para optimizaciones para servidor y para sistemas de escritorio.	10
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	La seguridad es uno de los principales objetivos de Debian. Todos los paquetes de seguridad clave incluyendo el paquete de memoria, núcleo y red por medio de un firewall que se actualizan diariamente. Por lo tanto, si un sistema se actualiza periódicamente (mediante el apt-get o dist-upgrade), la seguridad no es un tema para preocuparse.	10
18	**Estabilidad y Madurez**	Debian es una de las distribuciones más antiguas, tiene una gran estabilidad y la comunidad de desarrolladores, usuarios y simpatizantes, es muy amplia. Si se necesita un sistema maduro y bien probado, Debian es la opción correcta.	10
19	**Documentación**	Como parte de sus esfuerzos para crear un sistema operativo libre de gran calidad, el proyecto Debian está esforzándose en proporcionar a todos sus usuarios documentación adecuada en su propio idioma y accesible de manera sencilla como: manuales, procedimientos, pregunta frecuentes, documentos cortos, documentos históricos, etc.	10

Tabla 2- 3 Características técnicas de GNU/Linux Debian

2.2.4 OPENSUSE

Figura 2- 4 Símbolo de OpenSuse

2.2.4.1 Características Generales

El proyecto openSUSE es una comunidad mundial, patrocinada por Novell, con el objetivo de desarrollar un sistema Linux de escritorio amigable y fácil de usar. SUSE Linux, ha sido una de las principales distribuciones de Linux pues apareció por primera vez en 1992 como SuSE. Anteriormente era una solución para sistemas empresariales, ahora se concentra en usuarios de escritorio.

Últimamente ha ganado popularidad entre los usuarios de escritorio gracias a que ya se puede modificar su código.

OpenSUSE 10.1 fue el primer sistema autónomo con soporte de la tecnología innovadora desarrollada en Xgl de Novell. Xgl permite conseguir diferentes efectos a las estructuras visuales del escritorio dando características más suaves en tiempo real con el apoyo de una tarjeta de video sin tener mucha potencia de procesador.

El famoso YAST(acrónimo de Yet another Setup Tool, cuya traducción aproximada es "Otra Herramienta de Configuración Más") es una herramienta de instalación y configuración que se puede utilizar para configurar casi cualquier aspecto del sistema, como por ejemplo instalación de software, servicios de configuración, el uso compartido de archivos, configuración de los dispositivos externos, etc. YAST es considerado uno de los administradores todo en uno que integra paneles de control para GNU/Linux. Otra herramienta muy interesante es la herramienta de configuración gráfica X-Window - SaX2, la cual da la capacidad de elegir la tarjeta gráfica, sistema de resolución, profundidad de color, etc. Todo con unos pocos clics.

2.2.4.2 Características Técnicas

Ítem	Características	Detalles	Valoración Comparativa(*)
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390.	5
2	**Requisitos de Hardware Mínimos**	Modo texto: Procesador: 200 MHz Pentium, Memoria: 64MB, Disco Duro: 620MB. Modo gráfico: Procesador: 600 MHz Pentium, Memoria: 192MB, Disco Duro: 620MB.	7
3	**Licencia**	GNU GPL, desde febrero del 2006 Novell no incluye ningún driver propietario.	10
INSTALACIÓN			
4	**Instalador Global**	YAST es el instalador de SUSE Linux, permite a los usuarios instalar el sistema sin mayor problema, pero también configurar con mucho detalle las características durante la instalación.	10
5	**Selección de Paquetes**	Es una de las opciones permitidas por YAST durante la instalación, pudiendo ser descargados desde Internet o directamente desde los discos, la resolución de dependencias es automática.	10
6	**Grupo de paquetes predefinidos**	Los grupos de paquetes contienen: un sistema mínimo, sin sistema gráfico GNOME ni KDE, un sistema estándar con GNOME y con un sistema estándar de KDE.	4
7	**Instalación en Modo Experto**	El modo experto está disponible en la mayoría de las pantallas de instalación, es muy práctico y útil.	7
8	**Instalación Gráfica**	La instalación es completamente gráfica, personalizable y muy intuitiva gracias a la utilización de YAST.	10
9	**Velocidad de Instalación**	El tiempo de instalación es razonable. Se tarda de 15 a 25 minutos cuando las opciones se dejan por defecto.	9

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	YAST está muy bien diseñado y ofrece todas las funciones de una herramienta de configuración del sistema. El único inconveniente son los problemas de inestabilidad aleatoria.	10
11	**Manejo del Sistema Basado en Consola**	YAST también está disponible en modo consola, aunque con ciertas opciones limitadas.	5
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	SUSE tiene su propio repositorio de paquetes de alrededor de 14000 paquetes que utiliza YAST para actualizaciones e instalación de software. En general, el software certificado por Suse funciona sin problemas, pero los paquetes que no son certificados no son muy confiables.	8
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	YAST resuelve todas las dependencias automáticamente. Lamentablemente esto no se aplica a los paquetes no certificados.	8
14	**Herramientas Gráficas de Manejo de Paquetes**	YAST, la herramienta de configuración gráfica del sistema es irremplazable para procedimientos de instalación de software.	10
EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	La velocidad de arranque es lenta. Esta es una de las cuestiones que aun no recibe suficiente atención por parte de los desarrolladores de SUSE Linux con un promedio de 5 minutos.	1
16	**Velocidad de Respuesta del Sistema**	Es aceptable la velocidad y capacidad de respuesta, pero no existen optimizaciones especiales para escritorio ni para servidor.	1
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	Desde que SUSE fue adquirido por Novell, la seguridad se ha convertido en un aspecto muy importante y se ha realizado con AppArmor que es un paquete centrado en la seguridad, obligatorio para los programas de control de acceso a memoria, núcleo y protección contra la explotación de fallos.	10
18	**Estabilidad y Madurez**	Es un proyecto muy maduro, de código cerrado por un largo período de tiempo, pero en la actualidad es una distribución abierta,	5

		después de la estrategia global de Novell. El proyecto openSUSE es, sin embargo, más como un campo de pruebas para productos comerciales de SUSE y, por lo tanto, es por diseño un poco menos fiable que las versiones comerciales.	
19	**Documentación**	La localización del manual se ha simplificado gracias a YAST a partir de la versión SUSE 9.2, documentación adicional se puede encontrar en la página oficial, y en las comunidades que están aún en proceso de desarrollo.	5

Tabla 2- 4 Características técnicas de GNU/Linux Open Suse

2.2.5 SLACKWARE

Figura 2- 5 Símbolo de Slackware

2.2.5.1 Características Generales

Slackware es la más antigua de las distribuciones. La primera versión fue puesta en libertad el 13 de julio de 1993. El lema de Slackware es: "porque funciona" y que se dedica a "mantenerlo sencillo". Esta distribución es para usuarios avanzados ya que demanda conocimiento que se desarrolla básicamente en la línea de comandos.

Slackware es básicamente una distribución tradicional de los sistemas UNIX. Las jerarquías de directorios es clásica, contrariamente a otras distribuciones populares, Slackware contiene el núcleo original (no parchado), descargado directamente kernel.org y sin recompilar las funcionalidades.

La distribución no tiene una instalación gráfica, es instalado únicamente en modo texto, pero su instalador es muy intuitivo y no debería causar ningún tipo de problema. Cuenta únicamente con unos pocos de los asistentes de texto más básico para instalación de paquetes.

Existen dos líneas de Slackware: estable y actual. La versión actual se convierte en estable después de algunas versiones de prueba. Cada edición estable tiene su propio número. La línea actual no es versionada. Las versiones estables son ideales para servidores y equipos de escritorio conservadores.

El sistema de gestión de paquetes de Slackware se basa en simples paquetes .tgz[34] que no contienen toda la información sobre las dependencias. El usuario tiene que instalar manualmente todas las bibliotecas y los programas necesarios. La falta de gestión de las dependencias es a menudo mencionada como una debilidad Slackware, pero que indirectamente resuelve otro problema como es la dependencia infernal. Esta característica es Valoración Comparativa dada por los usuarios con experiencia, que son los principales seguidores y usuarios. Los sistemas de gestión de paquetes como rpm o dpkg están lejos de ser usados con facilidad ya que a veces, pueden causar problemas de dependencias redundantes, falta de dependencias, dependencias circulares y paquetes de conflicto.

Uno de los mayores inconvenientes en Slackware es el pequeño número de paquetes originales. Paquetes adicionales no oficiales se pueden encontrar en LinuxPackages Slacky.

2.2.5.2 **Características Técnicas**

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC.	6
2	**Requisitos de Hardware Mínimos**	Modo texto: Procesador: 200 MHz Pentium,	10

[34] Tgz extensión de archivo comprimido es el más comúnmente usado en sistemas operativos basados en UNIX.

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

		Memoria: 64MB, Disco Duro: 420MB.	
3	**Licencia**	GNU GPL en todos sus componentes.	10
INSTALACIÓN			
4	**Instalador Global**	A pesar de ser únicamente en modo texto, el instalador de Slackware no es muy difícil. Es bastante simple y bien pensado. Un usuario con conocimientos medios debería poder instalarlo sin mayores dificultades.	3
5	**Selección de Paquetes**	Existen disponibles varios modos de selección de paquetes: Pleno, en el que incluyen todos los paquetes de los grupos. Experto, en el que permite elegir paquetes individuales de los grupos seleccionados previamente. Novato, el que se instalan paquetes necesarios por defecto.	7
6	**Grupo de paquetes predefinidos**	Los paquetes están divididos en grupos en función de su finalidad: A (base), AP (aplicaciones más importantes), D (de desarrollo), E (Emacs), F (documentos), GNOME, K (código fuente), KDE, KDEI (Localización de KDE), L (bibliotecas), N (en red), T (Tetex / LaTeX), TCL, X (sistema de ventanas X), XAP (X apps) e Y (juegos).	10
7	**Instalación en Modo Experto**	Existe sólo en el modo Experto; sin embargo, hay modos de menú para la instalación de paquetes adicionales para cada modo.	10
8	**Instalación Gráfica**	La instalación es únicamente basada en consola.	1
9	**Velocidad de Instalación**	Velocidad promedio. Por defecto el sistema se instala en alrededor de 15 minutos.	10
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	No hay herramientas gráficas. Aunque se puede utilizar asistentes de los entornos: KDE, GNOME o XFCE.	1
11	**Manejo del Sistema Basado en Consola**	Vi, Emacs, son editores básicos, no cuenta con herramientas adicionales para el manejo del sistema.	1
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	Es muy pequeño el número de paquetes oficiales de alrededor de 1000 a 1500 paquetes. En la mayoría de los casos, será necesario el manual de compilación de los	4

		paquetes. Aun que existe una alternativa no oficial de paquetes Slackware que se puede descargar desde las páginas del proyecto o LinuxPackages y Slackyit.	
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	El sistema de gestión de paquetes se basa en simples paquetes tgz que no contiene toda la información sobre las dependencias.	1
14	**Herramientas Gráficas de Manejo de Paquetes**	No existe ninguna herramienta gráfica de gestión de paquetes oficial, pero existen herramientas no oficiales como: Smart Package Manager, GSlapt, XPKGTOOL, SlackMan,etc.	1
EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	La velocidad es aceptable, pero es muy fácil de optimizarla gracias a documentados tipo scripts, tiene un promedio de 1 minuto.	9
16	**Velocidad de Respuesta del Sistema**	A diferencia de Mandriva o Fedora Core, Slackware no tiene habilitados muchos servicios, por lo que la velocidad de respuesta del sistema es muy buena gracias a los scripts que permiten optimizar los servicios.	10
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	Posee una seguridad de alto nivel, con herramientas para protección de acceso a nivel de núcleo, memoria y red.	10
18	**Estabilidad y Madurez**	Patrick Volkerding, fundador de Slackware elige cuidadosamente los paquetes para tener la mejor estabilidad y fiabilidad de todas las distribuciones comparable con Debian.	10
19	**Documentación**	La localización de documentación, puede ser difícil para un usuario novato ya que las comunidades son escasas y la página oficial no contiene información completa.	2

Tabla 2- 5 Características técnicas de GNU/Linux Slackware

2.2.6 GENTOO

Figura 2- 6 Símbolo de Gentoo

2.2.6.1 **Características Generales**

Gentoo es una de las distribuciones más configurables y extensibles de los sistemas Linux. Permite al usuario decidir acerca de cada parte del sistema. Y gracias a la herramienta de gestión de software Portage[35] permite optimizar el sistema en su totalidad lo que aumenta la solidez del sistema operativo.

Gentoo tiene cierta diferencia a los habituales procesos de instalación de las distribuciones de Linux. Ya que permite seleccionar el núcleo, programas de registros de eventos y todas las características de bajo nivel de los sistemas operativos. Se necesita tiempo ya que no es tan fácil como "hacer clic", pero el beneficio es que tiene un control total sobre su sistema operativo.

Gentoo es conocido por su configurabilidad. Cada uno de los aspectos del sistema puede ser configurado por el usuario. Es el usuario quien decide si los programas son compilados con opciones seguras o algunas optimizaciones especiales. El inconveniente de este enfoque es que el usuario tiene que conocer muy bien su sistema antes de que razonablemente pueda configurar su sistema operativo.

Es una distribución para usuarios avanzados, conscientes y pacientes. Gentoo es una distribución que no depende de los paquetes binarios sino que más bien se concentra en la construcción del sistema operativo, con la meta de optimizar el hardware. La instalación y configuración de Gentoo puede tomar días o

---

[35] Portage, software para automatización de descargas desde los repositorios de Gentoo.

semanas. Los usuarios dicen que es porque vale la pena el control de la velocidad y las mejoras.

2.2.6.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC, ALPHA, MIPS, MIPSEL.	8
2	**Requisitos de Hardware Mínimos**	Modo texto: Procesador: 200 MHz Pentium, Memoria: 128MB, Disco Duro: 1.5 GB. Modo gráfico: Procesador: 600 MHz Pentium, Memoria: 256MB, Disco Duro: 3 GB.	7
3	**Licencia**	GNU GPL mayormente, pero incluye algunos controladores propietarios.	5
INSTALACIÓN			
4	**Instalador Global**	Dado que la distribución Gentoo 2006 viene con GNOME en su Live-CD, incluye también un instalador aunque no es lo suficientemente estable como para recomendar que los novatos lo usen. La instalación de Gentoo vía consola de comandos tarda unos días en los que se incluye la descarga de las fuentes, la compilación y la instalación, por lo que realmente no es una tarea para todos.	1
5	**Selección de Paquetes**	En el instalador gráfico, hay una breve lista de paquetes a elegir. Hay una opción para entrar en los paquetes requeridos, aunque esta tarea se realiza manualmente.	3
6	**Grupo de paquetes predefinidos**	No está disponible.	1
7	**Instalación en Modo Experto**	Existen manuales de instalación en los que se describen las diferentes etapas de instalación, ninguno de ellos es para usuarios	10

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

		novatos, aunque si son algo amigables.	
8	**Instalación Gráfica**	Es muy simple y está disponible en el Live-CD, pero es inestable.	7
9	**Velocidad de Instalación**	El manual de instalación de la fase 3 es algo más que un estándar de instalación de las distribuciones más populares. La instalación avanzada (etapas 1 y 2) es muy lenta y, por lo tanto, se recomienda sólo para los usuarios con mucha paciencia ya que puede llevar de 5 a 6 horas.	1
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	No se han desarrollado herramientas gráficas.	1
11	**Manejo del Sistema Basado en Consola**	Existe *vi*, pero no existen herramientas especializadas.	1
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	Existen alrededor de 11200 paquetes disponibles a través del sistema Portage, divididos en categorías claras.	8
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	Es muy buena la gestión de dependencia a través de Portage ya que permite instalar diferentes versiones de la misma aplicación con resolución de dependencias independientes.	8
14	**Herramientas Gráficas de Manejo de Paquetes**	Existen algunas aplicaciones gráficas, pero su utilización puede ser muy complicada. El trabajo basado en consola puede facilitar mucho el trabajo.	6
EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	Muy buena, se considera la mejor de todas las distribuciones con un tiempo de 30 a 40 segundos en instalaciones por defecto.	9
16	**Velocidad de Respuesta del Sistema**	Es un sistema muy sensible y es el resultado de la filosofía de Gentoo, pues provee paquetes optimizados para la arquitectura específica. Para lograr este objetivo, se debe tener algunos conocimientos para optimizarlo dependiendo del servicio.	5
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	El centro de seguridad es muy grande, se han endurecido los parámetros de acceso Web, núcleos (2.4 y 2.6). Gracias a GLSA -	4

		Consultiva de Seguridad de Gentoo Linux, no cuenta con herramientas de protección de acceso a memoria y red.	
18	**Estabilidad y Madurez**	Gentoo es una distribución activa y actualizada. Por lo tanto, algunos problemas pueden ocurrir con la inestabilidad.	5
19	**Documentación**	El grupo de Documentación de Gentoo Linux se encuadra dentro del proyecto de documentación Gentoo Linux que genera documentación en varios idiomas, la documentación es bastante completa.	10

Tabla 2- 6 Características técnicas de GNU/Linux Gentoo

2.2.7 MANDRIVA

Figura 2- 7 Símbolo de Mandriva

2.2.7.1 Características Generales

Mandriva Linux (antes Mandrake) se convirtió en una distribución muy popular entre los usuarios que han dejado de usar MS Windows[36], debido principalmente a su facilidad de uso. Mandriva es una distribución muy actualizada, y consiste en el software más reciente, que en ocasiones puede causar problemas de estabilidad.

Para uso de escritorio es aceptable para la mayoría de los usuarios y cuenta únicamente con una simple desventaja, ser muy actualizada, y la poca documentación que esto conlleva.

[36] MS Windows es una familia de sistemas operativos desarrollados y comercializados por Microsoft.

Mandriva es considerado como una de las mejores distribuciones Linux para novatos, junto con SUSE Linux y Ubuntu. Proporciona gran cantidad de asistentes útiles para la configuración gráfica de la mayor parte de las configuraciones del sistema.

Mandriva usa RPM como su formato de paquetes, pero no es compatible con los RPMs de Fedora o Suse. El gestor de paquetes por defecto es urpmi. El escritorio por defecto es KDE. También se incluye GNOME y configured. La última versión estable, Mandriva Discovery 2006, incluye algunas novedades como un firewall[37] interactivo y una herramienta de búsqueda KAT[38] (como Beagle o Google Desktop Search).

Mandriva puede ser instalado en todos los idiomas. Se ajusta perfecto para usuarios novatos, pero también es utilizado por una gran cantidad de usuarios con experiencia, que no gustan de luchar con Debian, Slackware o Gentoo.

2.2.7.2 **Características Técnicas**

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390.	5
2	**Requisitos de Hardware Mínimos**	Modo texto: Procesador: 166 MHz Pentium, Memoria: 64MB, Disco Duro: 620 MB. Modo gráfico: Procesador: 400 MHz Pentium, Memoria: 128 MB, Disco Duro: 620 MB.	10

[37] Firewall o cortafuegos es una parte de un sistema o una red que está diseñado para bloquear el acceso no autorizado.

[38] KAT, Gestor de búsqueda de ficheros.

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

3	**Licencia**	GNU GPL en su mayoría, pero incluye algunos controladores propietarios.	5
INSTALACIÓN			
4	**Instalador Global**	El instalador global es muy bueno y bien diseñado por su facilidad de acceder a configuraciones avanzadas como modo experto en la totalidad de las configuraciones, considerado como el más fácil de instalar de todas las distribuciones de Linux.	10
5	**Selección de Paquetes**	Está disponible como una de las opciones avanzadas durante la instalación, para ser obtenidos desde los discos o los repositorios vía Internet.	10
6	**Grupo de paquetes predefinidos**	Sabiamente se han determinado diferentes grupos de paquetes: 1) Trabajo (de escritorio, juegos, multimedia, Internet, herramientas de administración, programación). 2) Servidor (web, ftp, correo, base de datos, cortafuegos, enrutador). 3) Entornos de escritorio (KDE, GNOME, otros).	3
7	**Instalación en Modo Experto**	El modo experto es habilitado en cualquier momento del proceso de instalación.	10
8	**Instalación Gráfica**	Es muy intuitiva. Ofrece realmente muchas opciones de configuración básica. Todos ellos están bien descritos y las opciones por defecto son razonables.	10
9	**Velocidad de Instalación**	La instalación por defecto incluye muchos paquetes por lo que la instalación puede tomar algo de tiempo. Una instalación por defecto de Mandriva 2007 en un ordenador 1,6 GHz dura unos 25-30 minutos.	9
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	Contiene Lotes muy útiles con herramientas gráficas integradas en drakconf AKA "Mandriva Control Center (MCC)". Casi todos los aspectos del sistema operativo pueden ser configurados gráficamente.	5
11	**Manejo del Sistema Basado en Consola**	Últimamente muchos sistemas de administración de consola se han convertido en herramientas gráficas disponibles también en modalidad semi-gráficas. Es muy útil durante el uso de la consola remota ssh pero tiene sus limitaciones.	5

SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	Cuenta con un repositorio de paquetes muy grande alrededor de 16000 paquetes. Sólo Debian se compara con Mandriva en este aspecto. Hay una opción para añadir repositorios de lugares alternativos.	10
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	En la mayoría de los casos de instalación de software es tan fácil como entrar en urpmi y digitar el nombre del paquete. Los paquetes están bien descritos y fáciles de identificar. El único inconveniente es que se produzcan errores durante la instalación de dependencias.	8
14	**Herramientas Gráficas de Manejo de Paquetes**	Es muy conveniente en la pestaña del centro de control de Mandriva. La descarga de datos disponibles acerca de los paquetes puede tardar hasta dos minutos cuando utilizan todo el archivo hdlist.cz. Y significativamente más rápido con el archivo synthesis.hdlist.cz, que contiene menos información de paquetes, pero únicamente con paquetes certificados.	8
EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	La velocidad de arranque es rápida y comparable a la de Debian (de 40 a 50 segundos) debido a la utilización de pinit que es un script para la optimización de arranque utilizada desde el 2007.	9
16	**Velocidad de Respuesta del Sistema**	Tiene una velocidad promedio. Ya que Mandriva no se ha optimizado para dar una respuesta rápida, es el precio a pagar por la comodidad y mucha automatización.	1
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	Después de la instalación hay una opción para activar el servidor de seguridad y descargar los últimos parches. Es igualmente sencillo, sin embargo, la seguridad no es el principal objetivo de Mandriva ya que tienden a dar facilidad de uso, por lo tanto, no es tan seguro por defecto.	1
18	**Estabilidad y Madurez**	Mandriva por lo general se mantiene muy al día. Esto causa problemas de estabilidad.	5
19	**Documentación**	Cuenta con una amplia cantidad de comunidades en algunos idiomas, la página oficial cuenta con documentación y acceso a foros.	10

Tabla 2- 7 Características técnicas de GNU/Linux Mandriva

2.2.8 CENTOS

Figura 2- 8 Símbolo de Centos

2.2.8.1 **Características Generales**

CentOS (Community ENTerprise Operating System - Sistema Operativo de la Comunidad Empresarial) es una distribución Linux de clase empresarial derivada de los archivos fuentes provistos libremente al público por Red Hat. CentOS cumple completamente la política de redistribución y apunta a ser 100% compatible a nivel binario (programas) con Red Hat Enterprise Linux. En CentOS los principales cambios con respecto a RHEL(Ret Hat Enterprise Linux), es la eliminación de las ilustraciones y marcas de Red Hat de los paquetes. CentOS es gratuito, y está orientado a los usuarios que necesiten un sistema operativo de nivel empresarial, pero sin pagar los costos de certificación y soporte de Red Hat. CentOS es desarrollado por un creciente grupo de programadores. Los que son ayudados por una activa comunidad de usuarios, que incluye administradores de sistema, administradores de red, empresas, administradores, contribuidores del núcleo Linux y entusiastas de Linux de todo el mundo.

CentOS tiene algunas ventajas con respecto a proyectos similares: una activa y creciente comunidad de usuarios, desarrollo rápido, probado y corregido, una extendida red de réplicas, múltiples y gratuitas vías de soporte, foros, etc.

CentOS es un sistema estable que puede ser usado, ya sea como un servidor o como un sistema de escritorio de un usuario normal. Esto último requiere algunas modificaciones en la instalación por defecto.

2.2.8.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa*(*)
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC, ALPHA.	7
2	**Requisitos de Hardware Mínimos**	Modo texto: Procesador: 200 MHz Pentium, Memoria: 128MB, Disco Duro: 1 GB. Modo gráfico: Procesador: 400 MHz Pentium, Memoria: 512 MB, Disco Duro: 1 GB.	10
3	**Licencia**	Software GNU GPL en su totalidad.	10
INSTALACIÓN			
4	**Instalador Global**	El instalador es muy desarrollado, pues es similar a Red Hat utiliza Anaconda para realizar el proceso de instalación, ofrece funciones tanto para principiantes como para usuarios expertos. Contiene numerosas características que permiten personalizar la instalación del sistema operativo.	7
5	**Selección de Paquetes**	Los paquetes pueden ser seleccionados fácilmente e incluyen todas las dependencias, pero no describe muy claramente la utilidad de cada paquete.	7
6	**Grupo de paquetes predefinidos**	Esta característica es muy buena, pues, se realizó pensando en el perfil de uso del sistema operativo. Todos los grupos de paquetes incluyen paquetes instalados por defecto y opcionales. La instalación por defecto es un sistema de escritorio con GNOME, podrá elegir entre 4 modalidades: Escritorio Personal, Estación de Trabajo, Servidor o Personalizada.	4
7	**Instalación en Modo Experto**	Durante la instalación frecuentemente se puede ingresar a opciones de configuración avanzada, para realizar instalaciones no comunes.	7
8	**Instalación Gráfica**	La instalación gráfica utiliza Anaconda aunque se puede seleccionar la instalación basada en consola, ya que la instalación gráfica requiere de una elevada cantidad de memoria.	10

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

9	**Velocidad de Instalación**	El tiempo de instalación es algo extenso, aunque eso depende del perfil seleccionado, la instalación que demora más tiempo es el perfil de servidor, un sistema básico lleva alrededor de 2 horas y media.	4
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	Existen muchas herramientas que permiten la configuración en modo gráfico en su mayoría basado en GNOME. La mayor parte de configuración del sistema de operaciones se puede realizar sin necesidad de abrir la ventana del terminal.	5
11	**Manejo del Sistema Basado en Consola**	Centos contiene algunas herramientas de consola que permiten la gestión del equipo de manera integral, incluyen la configuración de la tarjeta de red, audio, video, servicios, etc.	10
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	El número de paquetes incluidos en los discos de instalación es muy completo aunque los repositorios no son tan extensos como los de sus competidores. Cuenta con repositorios activados: kbs-CentOS-Extras, update, rpmforge, base, contrib, addons, extras. El número de paquetes reportados por un listado con yum es 5785.	6
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	Posee un gestor de paquetes por defecto llamado yum, heredado de su fuente como es Red Hat Linux.	6
14	**Herramientas Gráficas de Manejo de Paquetes**	Centos 5 cuenta con herramientas gráficas basadas en yum como son: Pirut para realizar el manejo de paquetes y Pup para realizar las actualizaciones.	6
EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	El tiempo de arranque de Centos depende en gran parte de su configuración y de los servicios activados, si se configura adecuadamente se puede lograr un sistema de arranque muy rápido en promedio lleva de 1 a 2 minutos en una instalación por defecto.	7
16	**Velocidad de Respuesta del Sistema**	La velocidad de respuesta es bastante buena, aunque cuenta con configuraciones especiales para optimizaciones ya sea para uso de escritorio o de servidor que deben ser configuradas manualmente.	5

ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	Ofrece una gran cantidad de características de seguridad similares a Red Hat, partiendo de las aplicaciones más seguras como: permitir el acceso a nuevas herramientas de seguridad a nivel chip y memoria. La desventaja es que lleva un tipo considerable los parches oficiales para vulnerabilidades detectadas.	10
18	**Estabilidad y Madurez**	La estabilidad es una de las características más importantes de esta distribución, ya que se basa en paquetes de software bien probado y conservador que aseguran que su funcionamiento sea adecuado para ambientes empresariales.	10
19	**Documentación**	Existe disponible documentación muy variada, especialmente proveniente del proyecto Red Hat, y complementada por la comunidad de desarrolladores.	8

Tabla 2- 8 Características técnicas de GNU/Linux Centos

2.2.9 PCLINUXOS

Figura 2- 9 Símbolo de PCLinuxOS

2.2.9.1 Características Generales

PCLinuxOs es una distribución sin ánimo de lucro, basada inicialmente en MandrakeLinux. Es un solo CD auto arrancable con 2 GB de programas de escritorio y la capacidad de instalarse en el disco duro todo preparado para funcionar inmediatamente.

PCLinuxOS se distribuye en forma LiveCD, pero también puede ser instalado en el disco duro. La herramienta de instalación viene con una elegante herramienta de particionado. La operación en su conjunto no debe tardar más de 20 a 30 min.

PCLinuxOS utiliza APT (Advanced Packaging Tool – Herramientas Avanzadas de Empaquetamiento) para la gestión de software. Es muy probable que la mayor parte de los usuarios elijan sináptica para este efecto, ya que simplemente es una interfaz gráfica de usuario para APT. PCLinuxOS utiliza la tecnología de paquetes RPM, con un repositorio de cerca de 6500 paquetes.

El apoyo a la multimedia es un punto fuerte de PCLinuxOS. Soporta todos los formatos de archivo multimedia comunes. No tiene problemas para reproducir mp3, wma, ogg, mp4, avi, wmv, mov y archivos, sin la necesidad de instalar nuevos codecs. Todos los requisitos para el funcionamiento de los multimedia se cumplen por el sistema por defecto.

PCLinuxOS se distribuye con Java, Flash, y mplayerplug-in para multimedia Internet Firefox son soportados por el sistema.

2.2.9.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Arquitecturas Soportadas**	I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC.	6
2	**Requisitos de Hardware Mínimos**	Modo texto: Procesador: 400 MHz Pentium, Memoria: 192 MB, Disco Duro: 450 MB. Modo gráfico: Procesador: 600 MHz Pentium, Memoria: 256 MB, Disco Duro: 2 GB.	7
3	**Licencia**	GNU GPL en todos sus componentes.	10
INSTALACIÓN			
4	**Instalador Global**	PCLinuxOS se distribuye en forma LiveCD, pero también puede ser instalado en el disco duro (HDD). Se puede instalar en el disco duro a través de pasos muy sencillos aunque no es personalizable.	1

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.1 del ANEXO A.*

5	**Selección de Paquetes**	Se puede seleccionar de entre los paquetes que contiene el cdrom.	7
6	**Grupo de paquetes predefinidos**	Los paquetes son únicamente para KDE con varios entornos de escritorio disponibles a través de sináptica.	2
7	**Instalación en Modo Experto**	Disponible en varias ocasiones durante todo el proceso de instalación.	7
8	**Instalación Gráfica**	Contiene una versión bastante modificada de draklive-install. Que es uno de los instaladores de Linux más sencillo disponibles en la actualidad, por lo que no es personalizable.	3
9	**Velocidad de Instalación**	Lleva un tiempo promedio. La operación en su conjunto no debe tardar más de 20 a 30 min.	9
CONFIGURACIÓN			
10	**Manejo del Sistema Basado en Modo Gráfico**	Contiene una versión modificada del centro de control de Mandriva. La gran mayoría de las características de PCLinuxOS se puede configurar gráficamente.	5
11	**Manejo del Sistema Basado en Consola**	No existen herramientas que llamen la atención, únicamente vi.	1
SISTEMA DE PAQUETES			
12	**Cantidad de Paquetes**	PCLinuxOS viene en un Live CD con 2 GB de software comprimido en la imagen ISO. También hay más de 7000 paquetes disponibles en los repositorios PCLinuxOS.	6
13	**Gestión de Paquetes y Resolución Automática de Dependencias**	PCLinuxOS utiliza APT (Advanced Packaging Tool) para la gestión de software. Es muy probable que la mayor parte de los usuarios elijan sináptica para este trabajo. Ya que es únicamente una interfaz gráfica de usuario para APT. PCLOS utiliza la tecnología de paquetes RPM, y contiene un repositorio de cerca de 6500 paquetes de este tipo.	8
14	**Herramientas Gráficas de Manejo de Paquetes**	Contiene a sináptica para la descarga de paquetes y dependencias.	6
EFICIENCIA			
15	**Velocidad del Sistema de Arranque**	El arranque es en promedio de tres minutos para instalaciones por defecto.	5
16	**Velocidad de Respuesta del**	Optimizado para escritorio. Considerablemente mejor que Mandriva y	5

	Sistema	otras distribuciones que utilizan KDE.	
ESTABILIDAD Y DISPONIBILIDAD			
17	**Centro de Seguridad**	Lleva integrado el firewall Shorewall, que se pueden utilizar en un sistema de firewall dedicado, un multi-función gateway / router / servidor independiente o en un sistema GNU / Linux, pero no cuenta con protecciones a nivel de memoria y kernel.	4
18	**Estabilidad y Madurez**	Es muy estable y se adapta muy bien a hardware nuevo y antiguo ya que utiliza únicamente software comprobado.	10
19	**Documentación**	La documentación generada por las comunidades aún es reducida y en idioma Inglés únicamente.	4

Tabla 2- 9 Características técnicas de GNU/Linux Pclinuxos

2.3 ANÁLISIS DE SOFTWARE PARA LA IMPLEMENTACIÓN DE PROTOCOLOS DE ENRUTAMIENTO

2.3.1 FREESCO

2.3.1.1 Características Generales

FREESCO es una distribución de Linux tan pequeña como para ser almacenada en un diskette, esta propuesta nació como respuesta al costo de enrutadores Cisco, ya que intenta reemplazar a los de la gama baja, basándose en que los usuarios frecuentemente no usan todas las funcionalidades de estos equipos.

FREESCO proviene de *FREE cisco,* y aunque intenta reemplazar a los enrutadores Cisco, no se ha logrado esto por completo. La principal característica de FREESCO es su bajo costo con respecto a Cisco. La principal ventaja es que por su pequeño tamaño se lo puede implementar sobre un equipo antiguo y actualmente obsoleto como es un Pentium 386, pero una desventaja muy importante es no soportar enrutamiento dinámico sino únicamente enrutamiento estático.

Esta desventaja se basa en la proyección del proyecto FREESCO, ya que existe una gran cantidad de usuarios que no requieren del uso de protocolos más complejos y más bien requieren de un equipo que permita una configuración rápida y sin mayores complicaciones.

FREESCO soporta hasta 9 adaptadores Ethernet y hasta 2 módems, lo que permitiría el enrutamiento de una red de pequeñas y medianas empresas, a un costo muy bajo, una desventaja causada por el tamaño del Kernel es el limitado soporte de modelos de tarjetas de red.

Una ventaja importante de FREESCO es que se pueden levantar servicios adicionales al enrutamiento que se complementan con este trabajo, mejoran la seguridad de acceso y facilitan la administración de la red.

El procedimiento de instalación es muy rápido e incluye algunas configuraciones básicas de las cuales se puede elegir la que más se adapte a las necesidades del usuario y personalizarla agregando o eliminando elementos.

2.3.1.2 **Características Técnicas**

Ítem	*Características*	*Detalles*	*Valoración Comparativa*(*)
1	**Servicios adicionales**	• Gateway. • Firewall. • NAT (Traslación de Direcciones de Red -Network Address Translation). • Servidor DNS. • Servidor DHCP. • Servidor HTTP (público y de control). • Servidor Telnet (una conexión concurrente). • Servidor de Impresión.	10

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

		• Servidor de Acceso Remoto. • Servidor de Tiempo.	
2	**Licencia**	GNU GPL (Licencia Pública General).	5
3	**Modular**	No.	0
4	**Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	No maneja protocolos adicionales.	00
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	Únicamente rutas estáticas.	5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	No soporta.	0
REQUERIMIENTOS			
9	**Sistema operativo**	Es una distribución GNU LINUX de pequeño tamaño.	10
10	**Hardware**	CPU: 386 o superior. • Memoria RAM: Recomendado: 16 MB. • Floppy: 1.44 MB. • Disco Duro: No es necesario en sistemas con 8-16 MB RAM. • Adaptadores Ethernet: 3COM509, 3COM595, 3COM905, Realtek NE2000 compatible, Realtek NE2000 PCI compatible, ISA/PCI NE2000. • Módems: Únicamente módems externos.	10

Tabla 2- 10 Características técnicas de Freesco

2.3.2 BIRD

2.3.2.1 Características Generales

BIRD es un proyecto cuyo principal objetivo es el desarrollo de un completo demonio de “ruteo” de IP dinámica, en línea de comando, muy funcional y diseñado especialmente para ajustarse a cualquier sistema operativo basado en UNIX, con soporte para la mayoría de los protocolos existentes en Internet, como BGP, OSPF, RIP, así como sus variantes en IPv6.

BIRD brinda un mecanismo de configuración realmente flexible y fácil de maniobrar, basado en un fichero de configuración, que se activará mediante un SIGHUP (señales en consola).

BIRD ofrece un potente lenguaje para la aplicación de filtros de “ruteo” de IP dinámica, pero también de rutas estáticas, y con múltiples tablas de ruteo, ideal para dominar el conjunto de protocolos de “ruteo” vigentes en Internet.

Es un proyecto de software libre, es pequeño, rápido y funcional está disponible para la gran mayoría de distribuciones, es muy simple de compilar y configurar, el tamaño aproximado es de 198 KB por lo que se puede integrar en Puntos de Acceso Inalámbricos comerciales como Linuxap, Linksys, etc. Permite realizar scripts en los filtros, lo que facilita la depuración de errores, funciona muy bien con IPv6.

BIRD contiene una o más tablas de enrutamiento que pueden o no estar sincronizado con núcleo del sistema operativo, y que pueden o no estar sincronizadas entre sí. Cada tabla de enrutamiento contiene una lista de rutas conocidas.

2.3.2.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Servicios Adicionales**	No.	1
2	**Licencia**	GNU GPL (Licencia Pública General).	5
3	**Modular**	Si.	10
4	**¿Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	Si.	10
6	**¿Posee una versión de distribución pública y estable?**	Si	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	• BGP. • OSPF. • RIP. • Rutas estáticas.	5 5 5 5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	• BGP. • RIP. • Rutas estáticas.	5 5 5
REQUERIMIENTOS			
9	**Sistema Operativo**	GNU LINUX.	10
10	**Hardware**	Espacio en Disco: 500 KB. Memoria RAM: 64 MB (mínimo). Adaptadores Ethernet: 2 (mínimo).	7

Tabla 2- 11 Características técnicas de Bird

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

2.3.3 XORP

2.3.3.1 Características Generales

XORP es el acrónimo de eXtensible Open Enrutador Platform (Plataforma Extensible de Enrutador Libre), XORP es un software acogido en todo el mundo, con miles de descargas por parte de empresas, instituciones educativas y una activa comunidad de desarrolladores internacionales, su desarrollo se ha logrado gracias a la contribución de grandes empresas interesadas en el proyecto como: Intel, la Fundación Nacional de Ciencias, Google, Microsoft y Vyatta. XORP proporciona una plataforma completamente extensible ofreciendo protocolos de enrutamiento IPv4 e IPv6 con una plataforma unificada para configurarlos. Es la única plataforma de código abierto que ofrece capacidad integrada. La arquitectura de XORP es modular y permite una rápida introducción de nuevos protocolos, características y funciones.

XORP está disponible para su descarga gratuita bajo una licencia de tipo BSD (Berkeley Software Distribution), lo que permite la utilización del código fuente sin estar obligado a anunciar la procedencia del mismo, está escrito en C++ y se lo considera como uno de los proyectos de enrutamiento que plantea mayor expectativa por su flexibilidad, ya que puede ser implementado tanto en ambientes Linux, Mac OS y Windows Server 2003.

El equipo principal tiene su sede en el Instituto Internacional de Ciencias de Computación de Berkeley, California, XORP fue un proyecto fundado por Mark Handley en el año 2000, aunque su primera versión se liberó en el 2004, actualmente el proyecto está dirigido por Atanu Ghosh y la última versión es la 1.6.

XORP ha unificado una única interfaz de línea de comandos (CLI) que se utiliza para configurar los protocolos de enrutamiento y de las interfaces de red. La CLI de XORP se puede ampliar para abarcar otras funciones como manejo de cortafuegos, configuración NAT y DHCP. La arquitectura XORP permite también configurar diferentes protocolos de enrutamiento para correr con

diferentes seguridades, "sandboxes" ofrece la posibilidad de una mayor solidez y seguridad de las plataformas de enrutador alternativas.

XORP actualmente soporta versiones IPv4 e IPv6 de BGP4 +, OSPFv2, OSPFv3, RIP y RIPng para enrutamiento unicast, y PIM-SM y IGMP/MLD para la multidifusión. XORP funciona en la mayoría de Linux con kernel 2.4.x o 2.6.x y distribuciones BSD así como Mac OS X y Microsoft Windows.

2.3.3.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Servicios Adicionales**	• Firewall. • NAT (Network Address Translation).	4
2	**Licencia**	BSD (Berkeley Software Distribution).	10
3	**Modular**	Si.	10
4	**¿Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	Si.	10
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	• BGP. • OSPF. • RIP. • Rutas estáticas. • IGMP.	5 5 5 5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	• OSPF.	5

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

		• BGP. • RIP. • Rutas estáticas.	5 5 5
REQUERIMIENTOS			
9	**Sistema Operativo**	• GNU Linux. • Microsoft Windows 2003. • FreeBSD. • MacOS X.	10
10	**Hardware**	Espacio en Disco: 1.4 GB. Memoria RAM: 128 MB (mínimo). Adaptadores Ethernet: 2 (mínimo).	3

Tabla 2- 12 Características técnicas de XORP

2.3.4 ZEBRA – QUAGGA

2.3.4.1 Características Generales

Zebra - Quagga es un paquete de software de enrutamiento que proporciona enrutamiento basado en servicios de TCP/IP como RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, BGP-4 y BGP-4+. Quagga también soporta el comportamiento especial de BGP Route Reflector y Route Server. Además de los protocolos de encaminamiento tradicionales basados en IPv4, Zebra - Quagga también soporta protocolos de encaminamiento basados en IPv6. Soporta SNMP por medio del protocolo SMUX, Zebra - Quagga proporciona también las MIBs[39] para el monitoreo de los servicios.

[39] MIB, Management Information Base - Base de Información Gestionada, es un tipo de base de datos que contiene información jerárquica, estructurada en forma de árbol, de todos los dispositivos gestionados en una red de comunicaciones.

Quagga utiliza una arquitectura de software avanzada para proporcionar una gran calidad, con un motor multi-servidor de encaminamiento. Quagga tiene un interfaz de usuario muy interactivo para cada protocolo de enrutamiento, soporta también protocolos de enrutamiento basados en IPv6. Debido a su diseño modular es posible añadir protocolos adicionales.

Zebra - Quagga está distribuido bajo la Licencia General Pública GNU.

Tradicionalmente, la configuración de un enrutador basado en UNIX se realizaba mediante los comandos *ifconfig* y los comandos del tipo *route*. El estado de las tablas se podía mostrar mediante la utilidad *netstat*. Estos comandos solamente se podían utilizar trabajando como usuario root. Quagga, sin embargo tiene otro método de administración. En Quagga existen dos modos de usuario. Uno es el modo normal y el otro es el modo de habilitado (enable). El usuario de modo normal únicamente puede ver el estado del sistema, sin embargo el usuario de modo habilitado puede cambiar la configuración del sistema. Esta cuenta independiente de UNIX puede ser de gran ayuda para el administrador del enrutador.

Actualmente, Zebra - Quagga soporta los protocolos de unicast comunes, los protocolos de routing Multicast como BGMP, PIM-SM, PIM-DM están soportados en la versión 2.0. El soporte de MPLS está siendo programado actualmente. En el futuro, control de filtros TCP/IP, control de calidad de servicio QoS, la configuración de diffserv será añadida a Zebra. El objetivo de Zebra es conseguir un software de enrutamiento productivo de calidad y libre.

2.3.4.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Servicios Adicionales**	No.	1
2	**Licencia**	GNU GPL (Licencia Pública General).	5
3	**Modular**	Si.	10
4	**Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	Si.	10
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	• BGP. • OSPF. • RIP. • Rutas estáticas. • IGMP. • IS/IS.	5 5 5 5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	• OSPF. • RIP. • Rutas estáticas. • BGP.	5 5 5 5
REQUERIMIENTOS			
9	**Sistema Operativo**	• GNU Linux. • FreeBSD. • Solaris.	10

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

10	**Hardware**	Espacio en Disco: 2 GB. Memoria RAM: 256 MB (mínimo). Adaptadores Ethernet: 2 (mínimo).	1

Tabla 2- 13 Características técnicas de Zebra-Quagga

2.3.5 LEAF

2.3.5.1 Características Generales

Linux Embedded Appliance Firewall (LEAF – Aplicativo de Firewall Linux Embebido) es un software de enrutamiento personalizable con muy buenas características, que puede ser utilizado para una variedad de topología de red, a pesar de que es un software orientado al enrutamiento, LEAF puede ser utilizado de diversas formas, ya que es utilizado principalmente como gateway de Internet, firewall, y punto de acceso inalámbrico.

Es un proyecto que nace del proyecto LPR (Linux Router Proyect - Proyecto Enrutador Linux), muy similar a FREESCO y de igual manera es una distribución de código abierto por lo que se la puede utilizar sin la necesidad de adquirir ninguna licencia. Esta pequeña distribución tiene una funcionalidad tanto de enrutamiento como de firewall por lo que se volvió muy popular. Pero de igual manera que FREESCO solo se pueden implementar rutas estáticas, es una distribución de pequeño tamaño que puede correr sobre un CD. La última versión de LEAF es la 3.1 y fue liberada en marzo del 2007.

2.3.5.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa(*)*
1	**Servicios Adicionales**	• Punto de Acceso. • Gateway. • Firewall. • NAT (Network Address Translation).	7
2	**Licencia**	GNU GPL (Licencia Pública General).	5
3	**Modular**	No.	0
4	**¿Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	Si.	10
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	• Rutas estáticas.	5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	• No soporta.	0
REQUERIMIENTOS			
9	**Sistema Operativo**	Es una distribución GNU LINUX de pequeño tamaño.	10
10	**Hardware**	Espacio en Disco: 500 MB. Unidad de CD-ROM. Memoria RAM: 64 MB (mínimo). Adaptadores Ethernet: 2 (mínimo).	7

Tabla 2- 14 Características técnicas de Leaf

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

2.3.6 IPROUTE

2.3.6.1 Características Generales

IProute es una colección de herramientas (ifcfg, ip, rtmon y tc) para GNU/Linux que se utilizan para controlar el establecimiento de la red por medio del protocolo TCP/IP. Iproute es una herramienta software que proporciona a Linux un rendimiento y características con poca competencia en el panorama general de los sistemas operativos con características de enrutamiento. Este código de enrutamiento, filtrado y clasificación tiene más posibilidades que el que proporcionan muchos enrutadores, cortafuegos dedicados y productos de control de tráfico.

Iproute dispone de varias aplicaciones cuya funcionalidad es muy variada. Se puede hablar de comandos importantes como ip con los que se muestra o manipula el enrutamiento o los dispositivos de red, además de las políticas de encaminamiento y túneles que son posibles de implementar sobre esta herramienta. Además podemos encontrar aplicaciones como Traffic control (tc) que nos ofrece la posibilidad de mostrar y manipular el control del tráfico del enrutador.

Iproute aprovecha las capacidades del núcleo de Linux para controlar el flujo de transmisión, organizar prioridades del tráfico dependiendo de su naturaleza, establecer políticas de encolado, y eliminar paquetes, permite también realizar balanceo de carga asignando cantidades predefinidas de tráfico a cada interfaz del computador.

IProute cuenta con soporte para la mayoría de las tecnologías modernas de red incluyendo IP versiones 4 y 6, y nos permite realizar encapsulamiento de paquetes IPv6 en paquetes IPv4.

Una desventaja importante es la incapacidad de manejar más de un protocolo de enrutamiento dinámico, sino únicamente ARP, lo que lo hace muy útil para redes empresariales en los que el balanceo de carga y la calidad de servicio son un aspecto muy importante.

2.3.6.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa*(*)
1	**Servicios Adicionales**	• QoS. • Balanceo de Carga. • Firewall.	5
2	**Licencia**	GNU GPL (Licencia Pública General).	5
3	**Modular**	No.	0
4	**Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	No.	0
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	• Rutas estáticas. • ARP.	5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	• Rutas estáticas.	5
REQUERIMIENTOS			
9	**Sistema Operativo**	• GNU Linux.	10
10	**Hardware**	Espacio en Disco: 300 MB. Memoria RAM: 64 MB (mínimo). Adaptadores Ethernet: 3 (mínimo).	7

Tabla 2- 15 Características técnicas de Iproute

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

2.3.7 ROUTED

2.3.7.1 Características Generales

El demonio "routed" apareció con la finalidad de implementar el protocolo RIP (Routing Information Protocol). Sobre un sistema operativo para comunicar redes muy amplias, donde los vecinos son difíciles de identificar, soluciona también el problema del mantenimiento de ruta. Actualmente implementa RIPv1 y RIPv2 e Internet Enrutador Discovery Protocol para mantener las tablas de enrutamiento directamente en el núcleo del sistema operativo.

El demonio de enrutamiento routed se inicia automáticamente al iniciar los puertos de red activos y busca mensajes en cada uno de los dispositivos de red para poder determinar y poner al día la tabla de enrutamiento.

Para realizar el enrutamiento dinámico es necesario cumplir ciertas condiciones tales como:

- Que se tenga la posibilidad de elegir entre múltiples caminos para llegar a un destino.
- Que el demonio de enrutamiento dinámico pueda modificar automáticamente la tabla de enrutamiento para ajustarla a los cambios en la red.

Una desventaja de ROUTED es el de no poder manejar protocolos de enrutamiento para la versión 6 del protocolo IP.

2.3.7.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa*(*)
1	**Servicios Adicionales**	No posee servicios adicionales.	1
2	**Licencia**	GNU GPL (Licencia Pública General).	5

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

3	**Modular**	No.	0
4	**¿Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	No.	0
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	• RIP. • Enrutamiento Estático. • Internet Enrutador Discovery Protocol.	5 5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	No Soporta.	0
REQUERIMIENTOS			
9	**Sistema Operativo**	GNU Linux.	10
10	**Hardware**	Espacio en Disco: 30 MB. Memoria RAM: 128 MB (mínimo). Adaptadores Ethernet: 2 (mínimo).	3

Tabla 2- 16 Características técnicas de ROUTED

2.3.8 GATED

2.3.8.1 Características Generales

GATED, es un proyecto cuyo inicio se remonta a los primeros días de Internet con la fundación de NSFNet en 1987, que es unas de las marcas líderes en la industria del software de enrutamiento. El módulo de enrutamiento IPv6 ha sido desarrollado desde 1997. GATED también cuenta con el dominio de la primera parte del mercado de enrutamiento multicast, que ha estado en despliegue desde el año 1997. Enrutamiento y conmutación de software está disponible como fuente o como una completa solución para pequeñas y medianas empresas.

GATED NGC, es una completa solución de enrutamiento, ya que posee todos los protocolos de enrutamiento, organizados para su inclusión en una serie de enrutadores de la próxima generación. Dispone de componentes que incluyen: OSPF, BGP, IS-IS, DVMRP, PIM-SM, PIM-SSM, PIM-DM, MSDP, para IPv6 cuenta con: MP-BGP, IS-IS, Y Fast OSPF3. Que son productos ofrecidos a los fabricantes de equipos y se pueden adquirir las licencias por cada paquete o se puede adquirir una licencia que incluye una variedad de paquetes dirigidos a aplicaciones específicas.

La empresa NextHop distribuye, GATED ERS, que es una suite de software optimizado para memoria limitada en los sistemas integrados o para redes más pequeñas, y satisface las necesidades de los clientes, como fabricantes de equipos, empresas de hardware de seguridad, pequeñas oficinas domésticas, estos paquetes pueden soportar redes con IPv4 o IPv6.

2.3.8.2 Características Técnicas

Ítem	*Características*	*Detalles*	*Valoración Comparativa*(*)
1	**Servicios Adicionales**	Únicamente enrutamiento.	1
2	**Licencia**	Si.	0
3	**Modular**	Si.	10
4	**¿Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	Si.	10
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			
7	**Protocolos de enrutamiento**	• OSPF. • BGP.	5 5

(*)*La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

		• IS-IS. • DVMRP. • PIM-SM/SSM/DM. • MSDP.	
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos que soporta**	• MP-BGP. • OSPF3. • IS-IS.	5 5
REQUERIMIENTOS			
9	**Sistema Operativo**	GNU Linux.	10
10	**Hardware**	Espacio en Disco: 10 MB. Memoria RAM: 32 MB (mínimo). Adaptadores Ethernet: 2 (mínimo).	10

Tabla 2- 17 Características técnicas de GATED

2.3.9 COYOTE

2.3.9.1 Características Generales

Coyote Linux es una mini distribución gratuita de Linux que cabe en un disquete y requiere mínimos recursos de un computador, fue desarrollada por Vortech Consulting, su última versión es el 3.0.47 y se lanzó en abril del 2006.

Su función es habilitar un enrutador al que se conecta el módem adsl o cablemodem. También soporta dial up y conexión por la segunda placa de red a un hub o switch a las que se conectan los computadores que van a compartir Internet (hasta 256 computadores), asignando direcciones automáticamente por DHCP, también tiene un poderoso firewall , el hardware del computador no requiere monitor, teclado, ratón ni disco duro, lo que lo hace un sistema de muy bajo costo.

Sus características principales con las siguientes:

- Kernel 2.4.25.

- IpTables.

- Soporte para Ethernet (IP fija o dinámica), PPPoE y PPP DialUp.

- SSH 2.0.

- Soporte QoS.

Los requisitos a nivel de Hardware son escasos y los podemos suplir ampliamente con cualquier Pentium con más de 12 MB en RAM y una disquetera de 3 ½.

2.3.9.2 **Características Técnicas**

Ítem	*Características*	*Detalles*	*Valoración Comparativa*(*)
1	**Servicios Adicionales**	• Firewall. •NAT (Network Address Translation). • Servidor DHCP. • Servidor de Acceso Remoto. • Configuración de QoS.	8
2	**Licencia**	GNU GPL (Licencia Pública General).	5
3	**Modular**	No.	0
4	**¿Trabaja con el kernel?**	Si.	10
5	**¿Demonios independientes por cada protocolo?**	No.	0
6	**¿Posee una versión de distribución pública y estable?**	Si.	10
PROTOCOLO IP VERSIÓN 4 (IPV4)			

(*) *La valoración comparativa se asigna de acuerdo a la Tabla A.2 del ANEXO A.*

7	**Protocolos de enrutamiento**	Enrutamiento estático.	5
PROTOCOLO IP VERSIÓN 6 (IPV6)			
8	**Protocolos de enrutamiento**	No soporta.	0
REQUERIMIENTOS			
9	**Sistema Operativo**	Es una distribución GNU LINUX de pequeño tamaño.	10
10	**Hardware**	• Espacio en Disco: 500 KB. • Memoria RAM: 12 MB (mínimo). • Adaptadores Ethernet: 2 (mínimo). • Procesador: Pentium 486(mínimo). • Floppy: 1.44 MB.	10

Tabla 2- 18 Características técnicas de Coyote

2.4 ANÁLISIS DE SOFTWARE PARA LA IMPLEMENTACIÓN DE CALIDAD DE SERVICIO

Haciendo uso de las características que incluyen en el núcleo de Linux, es posible administrar el ancho de banda por medio del control de tráfico asignando prioridades a puertos, direcciones IP, rango de direcciones IP, protocolos, usuarios y horarios específicos, todo esto con algoritmos definidos para asignación de colas.

Traffic Shaping (TS) es una herramienta de software que permite administrar el recurso de acceso y ancho de banda de la red, es decir, realiza control de tráfico y da la posibilidad de asignar recursos en base a las necesidades de los usuarios y servicios de la red.

Traffic Shaping administra el proceso de asignación de colas y retardo en el envío de los paquetes, el proceso que realiza TS para la administración del tráfico es el siguiente:

Primero se analizan uno a uno los paquetes para determinar su origen, destino, protocolo y puertos. Luego, realiza la clasificación de los paquetes y a continuación se les asigna una cola dependiendo de la prioridad establecida para cada tipo de paquete.

Requerimientos	
Software	• Sistema Operativo GNU/LINUX. • Núcleo 2.6 (mínimo). • Iproute2. Ya que contiene la herramienta tc. • Iptables 1.3.5 o superior. • Parches para el núcleo e Iproute, soporte IMQ y ESFQ. • Layer7 herramienta para identificar protocolos P2P en base al contenido de los paquetes.
Hardware	• Espacio en Disco: 2 MB. • Memoria RAM: 128 MB. • Adaptadores Ethernet: 2 (mínimo).

Tabla 2- 19 Requisitos para software de Calidad de Servicio

2.4.1 SCRIPTS PARA LA IMPLEMENTACIÓN DE CALIDAD DE SERVICIO

En base a los algoritmos existentes para la implementación de Calidad de Servicio en sistemas Linux y a la complejidad de su implementación se han desarrollado ciertos scripts basados en los algoritmos que cuentan con mayor documentación y mejores características, para facilitar su implementación, los cuales se detallan a continuación.

2.4.1.1 CBQ-INIT

Este script se utiliza para simplificar la disposición y el manejo del control de tráfico de manera relativamente simple usando un sistema basado en CBQ en

Linux. El acceso a las características avanzadas de red del núcleo de Linux es proporcionado por la utilidades "ip" y "tc" del paquete de iproute2 de Kuznetsov, disponible en ftp://ftp.inr.ac.ru/ip-routing.

Dado que estas utilidades sirven específicamente para traducir deseos del usuario a comandos de RTNETLINK, su interfaz no es amigable, de difícil compresión y requiere la escritura de largas cadenas de comandos con parámetros. Esta tediosa tarea de escritura de comandos es lo que este script procura reducir.

El manejo de las facilidades avanzadas de una red en Linux es bastante flexible y este script trae algunas de estas características a los usuarios. Por supuesto, hay una relación inversamente proporcional entre la simplicidad y la flexibilidad. Si se considera que el script no permite la flexibilidad deseada a cambio de la simplicidad brindada, entonces puede realizar la configuración de prioridades directamente con la interfaz "ip" y "tc".

Si se desea que cqb.init traduzca sus configuraciones a comandos "tc", se puede usar el parámetro "compile" esto, generara los comandos "tc" requeridos para construir la configuración. Debe considerarse que "compile" no comprueba si los comandos "tc" son correctos, se utiliza esto solamente cuando el comando "startnocache", que es también útil al crear la configuración para comprobar si es totalmente válido. Se debe tener en cuenta que todos los parámetros de CBQ son válidos para los interfaces ethernet únicamente.

2.4.1.2 **WONDERSHAPER**

Es un script muy sencillo que nos definirá las colas necesarias para implementar calidad de servicio basado en el algoritmo HTB Hierarchical Token Bucket – Marcado Jerárquico de Paquetes.

Previamente a la implementación del script, necesitamos habilitar ciertas características del núcleo que debe ser por lo menos la versión 2.4 pero se recomienda una superior.

Se debe tener al menos las siguientes opciones activadas dentro del núcleo de Linux: "Networking Options" / "QoS and/or Fair Queueing": CBQ, PRIO, SFQ, Qdisc, QoS, Rate Estimator, Packet classifier, Firewall based classifier. U32 classifier y Traffic Policing. Aunque es recomendable utilizar todas la opciones disponibles dentro de QoS y Fair Queueing.

También debemos instalar el paquete iproute más reciente.

Wonder Shaper nos proporciona wshaper.htb que usa el algoritmo HTB debe tener en cuenta que HTB no viene incluido en todos los núcleos 2.4, ya que incluye a partir de la versión 2.4.21.

Para iniciar la implementación se debe comprobar que la versión del algoritmo HTB del núcleo corresponda con la versión del binario tc que viene en el paquete iproute. Si al ejecutar el script wshaper.htb se obtienen errores y en el syslog nos denota un mensaje acera de la incompatibilidad de la versión del HTB, se debe buscar una versión más reciente de iproute.

Una vez listo el núcleo y el iproute adecuado e instalado, pasamos al script que define las colas.

En el script propiamente se debe definir los anchos de banda de bajada y de subida y el interfaz de red donde está conectado el enrutamiento. Modificando las líneas en las que se definen la capacidad de los enlaces y se asigna las prioridades a los diferentes dispositivos y puertos.

En el script wshaper.htb puede personalizarse varias características, que permitirán configurar las opciones de calidad de servicio en base a las necesidades de los usuarios y las características de la red donde será implementada.

2.5 ANÁLISIS DE SOFTWARE PARA LA IMPLEMENTACIÓN DE SEGURIDAD

2.5.1 STRONGSWAN

StrongSwan es una aplicación completa de IPsec para los núcleos 2.4 y 2.6 de Linux.

StrongSwan es un descendiente del proyecto FreeS / Wan, y es distribuido bajo licencia GPL. StrongSwan es un proyecto impulsado por Andreas Steffen que es profesor para la Seguridad en las Comunicaciones en la Universidad de Ciencias Aplicadas en Rapperswil Suiza. El enfoque del proyecto tiene su fortaleza en los mecanismos de autenticación X.509 utilizando certificados de llave pública, adicionalmente permite el almacenamiento seguro de claves privadas en tarjetas inteligentes a través de una normalización de interfaz PKCS # 11, como una característica opcional. Soporta listas de revocación de certificados y el Protocolo de Estado de Certificados en Línea, Online Certificate Status Protocol (OCSP). Una característica única es el uso de certificados X.509 como atributo para aplicar sistemas de control avanzados de acceso basado en grupos.

StrongSwan posee un amigable y sencillo enfoque para la configuración e interopera sin problemas con la mayoría de las otras implementaciones IPsec, incluidas las de Microsoft Windows y Mac OS X.

El diseño modular StrongSwan 4.2 es una versión que cumple completamente el protocolo IKEv2 definido en el RFC 4306. El arquitecto del software y principal desarrollador del demonio IKEv2 es Willi Martin.

NAT trasversal para IKEv2 ha sido aportada por Tobias Brunner y Daniel Röthlisberger.

2.5.1.1 **Requerimientos**

- Kernel de Linux, ya sea 2.0, 2.2, 2.4 o 2.6.
- Bibliotecas libgmp.

2.5.1.2 **Características Técnicas**

Corre tanto en núcleos Linux 2.4 (Klips) y Linux 2.6 (IPsec nativo).

Posee cifrado 3DES, AES, Serpent, Twofish, Blowfish.

Implementa los protocolos de intercambio IKEv1 e IKEv2 (RFC 4306).

Soporte comprobado de túneles IPv6 para conexiones IPsec.

Soporte de direcciones IP dinámicas e interfaz de actualización con IKEv2 MOBIKE (RFC 4555).

Conexión, puesta en marcha y actualización periódica.

Inserción automática y supresión de políticas IPsec basadas en reglas de firewall.

NAT-Trasversal UDP a través de encapsulación y puerto flotante (RFC 3947).

Servidor XAUTH, funcionalidad de cliente IKE principal para modo de autenticación.

Detección Dead Peer (DPD, RFC 3706) se ocupa del cierre de los túneles.

La autenticación basada en certificados X.509 o llaves pre-compartidas.

Generación por defecto de certificado auto firmado durante el primer arranque strongSwan.

Recuperación y almacenamiento local de listas de revocación de certificados a través de HTTP o LDAP.

Soporte completo de Online Certificate Status Protocol (OCSP, RCF 2560).

CA gestión (OCSP y CRL URI, por defecto servidor LDAP).

Potentes políticas IPsec basadas en comodines o AC intermedios.

Grupo de políticas basadas en atributos de certificados X.509 (RFC 3281).

Almacenamiento opcional de llaves privadas RSA y certificados en una tarjeta inteligente.

Acceso normalizado Smartcard a través de interfaz PKCS # 11.

PKCS # 11 función de descifrado RSA proxy que ofrecen servicios a través de whack.

StrongSwan Manager - una interfaz de administración gráfica para IKEv2.

2.5.2 IPSEC-TOOL RACOON

IPsec-Tools es una extensión para Linux 2.6 de las utilidades KAME de BSD.

Racoon es una implementación Open Source de IPsec para el sistema operativo Linux. Se trata de un código proveniente del proyecto KAME. Racoon es un esfuerzo conjunto para crear un sólido y único conjunto de software, especialmente dirigidas a IPv6/IPsec. Talentosos investigadores de varias organizaciones japonesas se sumaron al proyecto. Este esfuerzo conjunto permite evitar la duplicación de desarrollo en la misma zona, de manera eficaz y proporcionar una alta calidad y avanzadas funciones de paquete.

- Incluye ipsec-tools y racoon (que permite negociar las claves automáticamente).
- Ipsec-tools incluye el comando setkey, que permite modificar tanto la bases SAD y SPD para realizar una configuración manual.
- Utilizando racoon se puede utilizar el protocolo IKE para realizar el negociado de SAs, lo que nos permite configuraciones más escalables. No obstante, se deben establecer de igual modo las políticas SP mediante setkey. En el caso de racoon la autenticación entre distintos nodos se realiza mediante certificados X.509, kerberos o claves pre compartidas.

2.5.2.1 **Requerimientos**

- Núcleo de Linux 2.6.
- Bibliotecas libgmp.

2.5.2.2 **Características Técnicas**

Soporte de Certificados X.509.

Soporte por defecto de llaves RSA.

Soporte de llaves RSA desde DNS.

Integra inicio rápido de las políticas IPsec.

OpenSC es la interfaz para uso de smartcard.

Uso de L2TP para múltiples clientes tras el mismo router NAT.

Tiene la capacidad de conexión con sistemas Windows y Mac OSX.

Soporta XAUTH, para modo de autenticación vía PAM y archivo de *passwd*.

Soporte de DNSSEG.

Proceso de ayuda y encriptado mediante Pluto.

2.6 SELECCIÓN DE LA DISTRIBUCIÓN EN BASE A LA NORMA IEEE 830

2.6.1 ESPECIFICACIÓN DE REQUISITOS SOFTWARE

2.6.1.1 **Introducción**

Este documento es una Especificación de Requisitos Software (ERS) para el sistema operativo GNU / Linux de propósito general, en cuya plataforma será implementado un ruteador dual IPv4/IPv6, esta distribución va a permitir la configuración de su Kernel e implementar paquetes para realizar enrutamiento dinámico, paquetes para gestionar calidad de servicio y seguridad por medio del protocolo IPsec. Esta especificación se ha estructurado en base a los criterios planteados en el plan de tesis del proyecto.

2.6.1.2 **Propósito**

El objetivo del uso de estas especificaciones es definir claramente y de forma precisa todas las funcionalidades y restricciones del sistema operativo utilizado en el presente proyecto. El documento va dirigido hacia los implementadores del proyecto, el tribunal de revisión y la comunidad de posibles usuarios finales del producto, como un documento de respaldo de un diseño robusto y confiable.

2.6.1.3 **Ámbito del sistema**

Basados en las variadas funcionalidades que el sistema operativo Linux posee se desea implementar un sistema estable y robusto basado únicamente en software libre para facilitar el trabajo de enrutamiento en lugares donde no se manejen redes complejas pero que requieran ciertas funcionalidades que permitan un mejor desempeño de la red.

Se ha constatado que bajo Linux se puede implementar un ruteador que soporte tanto datagramas IPv4 e IPv6, que responda eficazmente a los problemas de seguridad observados en el Internet gracias a la configuración de IPsec, y que realice un control eficaz de tráfico dentro de la red.

El sistema operativo debe ser estable, maduro y con un gran repositorio de paquetes de software libre que puede ser configurado dentro de su plataforma.

2.6.1.4 **Acrónimos**

GNU

El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

ERS

Documento de Especificación de Requisitos Software.

GFDL

Licencia para Documentación Libre de GNU.

GPL

Licencia para Software Libre de GNU.

2.6.1.5 **Referencias**

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998. Adjunta como **ANEXO F**.
- Proyecto de Titulación: Análisis, implementación y evaluación de un prototipo ruteador dual Ipv4/Ipv6 con soporte de QoS e IPsec sobre Linux, usando AHP para la selección del hardware e IEEE 830 para la selección del software. Capítulo 2 Análisis de Software para la implementación de los protocolos. Pazmiño C, Jiménez G, 2008.

2.6.1.6 **Visión general del documento**

En este documento especificamos tres secciones, en la primera damos una introducción en la cual tenemos una visión general de un ERS. En la siguiente sección describimos características que debe cumplir el sistema operativo, las restricciones, las dependencias que afectan al desarrollo del enrutador. En la tercera sección se definen detalladamente los requisitos que debe satisfacer el sistema.

2.6.1.7 **Descripción general**

2.6.1.7.1 Perspectiva del producto

El sistema operativo permitirá una correcta configuración, administración y monitoreo del enrutador, interactuará con los clientes mediante una consola de administración.

2.6.1.7.2 Funciones del sistema

2.6.1.7.2.1 Proveer un ambiente seguro y estable

Que permita la instalación, configuración e implementación de herramientas de software para enrutamiento, calidad de servicio e IPsec.

2.6.1.7.2.2 Facilitar el uso del núcleo 2.6 de Linux

Para permitir el aprovechamiento de las características de enrutamiento, calidad de servicio y seguridad IP, ya que el núcleo 2.6 incluye estas características por defecto y no es necesario parchar el núcleo.

2.6.1.7.2.3 Manejo adecuado de software

Permitir el manejo adecuado y eficiente de todo el software, interfaces, sistema operativo y protocolos de enrutamiento.

2.6.1.7.3 Restricciones

El sistema operativo será software libre (de acuerdo con la licencia GNU-GPL o similar) y deberán ser libres aquellos componentes que reutilice.

El sistema operativo deberá trabajar por medio de interfaces ethernet.

2.6.1.7.4 Suposiciones y dependencias

2.6.1.7.4.1 Suposiciones

Se asume que el sistema operativo tendrá un comportamiento estable, que permitirá trabajar sin ningún inconveniente al resto de paquetes implementados sobre él.

2.6.1.7.4.2 Dependencias

El sistema operativo dependerá exclusivamente de una adecuada reconfiguración del núcleo, que nos permita obtener un ambiente propicio para que el software de enrutamiento realice su trabajo eficientemente.

2.6.1.7.4.3 Requisitos específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son esenciales. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio estabilidad, disponibilidad y confiabilidad.

2.6.1.7.5 Requisitos funcionales

Código	Ítem	Descripción
REQ01	Software Libre	El sistema operativo deberá tener licencia GPL o una licencia similar sin costo.
REQ02	Poseer una versión estable	El software no debe ser una versión de prueba ni versiones en desarrollo, es decir debe contar con versiones estables distribuidas públicamente.
REQ03	Seguridad	Poseer un nivel de seguridad, que garantice un acceso restringido al núcleo de memoria, protección, detección y reordenamiento de desbordamiento de memoria.
REQ04	Eficiencia	La velocidad del sistema de arranque debe ser lo más rápido posible, para obtener un tiempo de convergencia muy pequeño con el objetivo de mantener una red de comunicaciones eficiente.
REQ05	Velocidad de Respuesta del Sistema	El sistema operativo debe tener una respuesta rápida a la transferencia de información y reenvío de paquetes.
REQ06	Documentación y Ayuda	Tener una amplia y disponible documentación, de parte de los desarrolladores del sistema operativo como de la comunidad de Internet, para facilitar el aprendizaje y conocimiento.

REQ07	Amplio número de paquetes	Disponer de los paquetes necesarios para la implementación de protocolos de enrutamiento, calidad de servicio e IPsec o sus dependencias, que estén incluidos en sus repositorios oficiales y tengan licencia GNU GPL.
Requisitos de interfaces externos		
	Interfaces de usuario	
REQ08	Configurabilidad	El sistema operativo contará con facilidades para que los administradores puedan configurar todas las funcionalidades y funciones por medio de herramientas basadas en consola de texto.

Tabla 2- 20 Requisitos funcionales para selección de sistema operativo

2.6.1.8 Selección

Para la referencia se consideró la fila de valoración comparativa de las tablas 2-1 a 2-9 de características técnicas de los sistemas Operativos

Selección de Distribución GNU/Linux										
CODIGO	REFERENCIA	FEDORA	UBUNTU	DEBIAN	OPENSUSE	SLACKWARE	GENTOO	MANDRIVA	CENTOS	PCLINUXOS
REQ01	Item 3	5	5	10	10	10	5	5	10	10
REQ02	Item 18	5	5	10	5	10	5	5	10	10
REQ03	Item 17	10	8	10	10	10	4	1	10	4
REQ04	Item 15	5	8	9	1	9	9	9	7	5
REQ05	Item 16	5	5	10	1	10	5	1	5	5
REQ06	Item 19	10	7	10	5	2	10	10	8	4
REQ07	Item 12	8	8	10	8	4	8	10	6	6
REQ08	Item 11	10	10	10	5	1	1	5	10	1
TOTAL		58	56	79	45	56	47	46	66	45

Tabla 2- 21 Selección de Sistema Operativo

Obteniendo la mayor valoración para la distribución Debian.

2.7 SELECCIÓN DE SOFTWARE DE ENRUTAMIENTO BASE A LA NORMA IEEE 830

2.7.1 ESPECIFICACIÓN DE REQUISITOS SOFTWARE

2.7.1.1 Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el subsistema de enrutamiento que será levantado sobre el sistema operativo Linux. Este software deberá contar con las características necesarias para efectuar el mismo trabajo realizado por un enrutador basado en hardware proyectándose como una solución modular, innovadora y de menor costo que un equipo con características similares.

Esta Especificación Requisitos Software (ERS) para el sistema operativo GNU/Linux de propósito general, en cuya plataforma será implementado un ruteador dual IPv4/IPv6, esta distribución va a permitir la configuración de su Kernel e implementar paquetes para realizar enrutamiento dinámico, paquetes para gestionar calidad de servicio y seguridad por medio del protocolo IPsec. Esta especificación se ha estructurado en base a los criterios planteados en el plan de tesis del proyecto.

2.7.1.1.1 Propósito

El objetivo del uso de estas especificaciones es definir claramente y de forma precisa todas las funcionalidades y restricciones del software encargado del enrutamiento que se desea implementar en el presente proyecto.

El documento va dirigido hacia los implementadores del proyecto, el tribunal de revisión y la comunidad de posibles usuarios finales del producto, como un documento de respaldo de un diseño robusto y confiable.

2.7.1.1.2 Ámbito del sistema

El sistema nace por los elevados costos que representa para las pequeñas y medianas empresas la adquisición de equipos de enrutamiento confiables y que permitan su adaptación a las nuevas tecnologías sin tener que invertir elevados costos para la renovación tecnológica.

El software de enrutamiento debe ser de libre distribución, pues es un objetivo del proyecto la no utilización de software que requiera de licencias con costo, técnicamente el software debe ser modular, robusto y principalmente estable.

El software debe soportar protocolos de enrutamiento estáticos y dinámicos para las dos versiones de protocolo IP existentes (IPv4 e IPv6).

Entre los protocolos de enrutamiento dinámico que deberá soportar son: OSPF, RIP y BGP es sus versiones más recientes para IPv4 e IPv6.

El ámbito del equipo desarrollado integra la implementación y prueba del software de enrutamiento en ambientes reales donde se pueda evaluar el desempeño y la funcionalidad, para luego compararlo con equipo que tengan características similares de funcionamiento, con los que se realizará una comparación financiera que nos permita establecer la conveniencia económica del mismo.

2.7.1.1.3 Acrónimos

GPL

Licencia Publica General (Software Libre de GNU).

OSPF

Open Shortest Path First (Abre Primero Ruta Más Corta)

Algoritmo de estado de enlace de enrutamiento jerárquico propuesto como sucesor del RIP en la comunidad de la Internet. Entre las características de OSPF se incluyen enrutamiento más económico, enrutamiento multiruta y equilibrio de carga. OSPF es un derivado de una versión anterior del protocolo IS-IS.

RIP

Routing Information Protocol (Protocolo de Información de Rutas).

Protocolo del encaminamiento directivo de información en una área extensa de la red.

BGP

Border Gateway Protocol (Protocolo de Entrada Límite).

Protocolo usado para intercambiar vías de información entre entradas de servidores en redes de gran extensión como el Internet.

IPv4

Versión 4 del Protocolo IP (Internet Protocol).

IPv4 usa direcciones de 32 bits, limitándola a 2^{32} = 4.294.967.296 direcciones únicas, muchas de las cuales están dedicadas a redes locales (LANs). Por el crecimiento enorme que ha tenido del Internet (mucho más de lo que esperaba cuando se diseñó IPv4), combinado con el hecho de que hay desperdicio de direcciones en muchos casos.

IPv6

Versión 6 del Protocolo IP (Internet Protocol).

Es un estándar en desarrollo del nivel de red encargado de dirigir y encaminar los paquetes a través de una red. Las direcciones IPv6, definidas en el RFC 2373 y RFC 2374, son de 128 bits; esto corresponde a 32 dígitos hexadecimales, que se utilizan normalmente para escribir las direcciones IPv6.

2.7.1.1.4 Referencias

IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.

Proyecto de Titulación: Análisis, implementación y evaluación de un prototipo enrutador dual Ipv4/Ipv6 con soporte de QoS e IPsec sobre Linux, usando AHP para la selección del hardware e IEEE 830 para la selección del software.

Capitulo 2 Análisis de Software para la implementación de los protocolos. Pazmiño C, Jiménez G, 2008.

2.7.1.2 **Descripción General**

2.7.1.2.1 Perspectiva del producto

El software de enrutamiento realizará un correcto reenvío de paquetes entre redes, por medio de uno o varios de los protocolos de enrutamiento implementados, permitirá también utilizar políticas de calidad de servicio para optimizar el uso del ancho de banda en los enlaces, adicionalmente el software deberá tener la capacidad de levantar varios protocolos de enrutamiento simultáneamente y como servicios independientes.

2.7.1.2.2 Funciones del Sistema

2.7.1.2.2.1 Reenviar los paquetes

Recibir los paquetes por las diferentes interfaces de red y reenviarlos a los distintos destinos.

2.7.1.2.2.2 Administrar los protocolos de enrutamiento

Permite definir las directivas de enrutamiento para los distintos protocolos en cada versión del Protocolo IP.

2.7.1.2.2.3 Configurar los protocolos de enrutamiento

Facilitar los comandos y el ambiente propicio para realizar las distintas configuraciones necesarias para el levantamiento de los servicios de enrutamiento.

2.7.1.2.2.4 Modular

El software deberá tener la capacidad de levantar varios protocolos de enrutamiento simultáneamente y con demonios independientes.

2.7.1.2.3 Restricciones

El sistema será software libre (de acuerdo con la licencia GNU-GPL o similar) y deberán ser libres aquellos componentes que reutilice.

El sistema basará sus comunicaciones en protocolos estándar de Internet.

El software deberá trabajar por medio de interfaces Ethernet para realizar el trabajo de enrutamiento.

2.7.1.2.4 Suposiciones y dependencias

2.7.1.2.4.1 Suposiciones

Se asume que el software de enrutamiento tendrá un comportamiento estable luego de ser implementado sobre el sistema operativo, y se realizará las pruebas pertinentes de desempeño para probar esta suposición.

2.7.1.2.4.2 Dependencias

El software de enrutamiento dependerá directamente de la estabilidad del sistema operativo.

Para enrutamiento dinámico dependerá también de enrutadores vecinos que le permita actualizar sus tablas de enrutamiento.

2.7.1.2.4.3 Requisitos específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son esenciales,

Estos requisitos se han especificado teniendo en cuenta, entre otros, los criterios de estabilidad, disponibilidad y confiabilidad.

2.7.1.2.5 Requisitos funcionales

Código	Ítem	Descripción
REQ01	RIP	El software deber soportar las versiones de RIP Versión 1, Versión 2, para Ipv4 y RIPng para IPv6.
REQ02	OSPF	El software deber soportar las versiones de OSPF Versión 2, para Ipv4 y Versión 3 para IPv6.
REQ03	BGP	El software deber soportar las versiones de BGP Versión 4, para Ipv4 y MPBGP para IPv6.
REQ04	Enrutamiento Estático	El software debe permitir configurar rutas estáticas
REQ05	Servicios Independientes	Los protocolos deberán poder levantarse independientemente aunque cualquiera de los protocolos restantes estén activos, es decir podrán trabajar simultáneamente.
REQ06	Compatible con sistema operativo GNU/Linux	El software debe ser compatible con sistemas operativos basados en Unix, específicamente GNU/Linux.
REQ07	Poseer una versión estable	El software no debe ser una versión de prueba ni versiones en desarrollo, es decir debe contar con versiones estables distribuidas públicamente.
Requisitos de interfaces externos		
	Interfaces hardware	
REQ08	Interfaces Ethernet	Tres interfaces Ethernet, destinados a la Intranet, a la WAN, y una para las direcciones que se publicaran a Internet.
	Requisitos de rendimiento	
REQ09	Modular	El software deberá levantar servicios de enrutamiento de protocolos adiciones sin necesidad

		de suspender ni retardar los que están levantados.
	Atributos	
REQ10	Software libre	El sistema será software libre y, por tanto, cualquier componente software que reutilice también deberá ser libre.

Tabla 2- 22 Requisitos funcionales para selección de software de enrutamiento

2.7.1.2.6 Selección

Para la Referencia se consideró la fila de valoración comparativa de las tablas 2-10 a 2-18 de características técnicas del software de enrutamiento.

Selección de Software de Enrutamiento										
CODIGO	REFERENCIA	FREESCO	BIRD	XORP	ZEBRA	LEAF	IPROUTE	ROUTED	GATED	COYOTE
REQ01	Item 7 y 8	0	10	10	10	0	0	0	10	0
REQ02	Item 7 y 8	0	5	10	10	0	0	0	10	0
REQ03	Item 7 y 8	0	10	10	10	0	5	5	10	0
REQ04	Item 7 y 8	5	10	10	10	5	5	5	10	5
REQ05	Item 5	0	10	10	10	10	0	0	10	0
REQ06	Item 9	10	10	10	10	10	10	10	10	10
REQ07	Item 6	10	10	10	10	10	10	10	10	10
REQ08	Item 10	10	7	10	1	7	7	3	10	10
REQ09	Item 3	0	10	3	10	0	0	0	10	0
REQ10	Item 2	5	5	10	5	5	5	5	0	5
TOTAL		40	87	93	86	47	42	38	90	40

Tabla 2- 23 Selección de Software de Enrutamiento

Obteniendo la mayor valoración para el software de enrutamiento XORP.

CAPÍTULO 3. IMPLEMENTACIÓN DE PROTOTIPO

3.1 INTRODUCCIÓN

En el presente capítulo se realizó un estudio para la selección de hardware mediante el Proceso de Análisis Jerárquico AHP, con una muestra de 29 encuestas se obtuvo una tabla final de resultados que nos permite apreciar el aspecto con mayor prioridad y por ende el cual consideramos para la compra del hardware.

Luego se describen dos ambientes en los cuales será implementado el enrutador, se procede a la implementación y evaluación de las características de desempeño del enrutador dual, con los conocimientos adquiridos y detallados en los anteriores capítulos respecto a los protocolos de enrutamiento y al protocolo IP en sus dos versiones, los conocimientos del sistema operativo Linux específicamente hablando de la distribución seleccionada en base a la norma IEEE830 Debian, conocimientos de calidad de servicio y del estándar de seguridad para IP; se implementará un prototipo estándar que permita funcionar de manera óptima en los ambientes de trabajo reales, y se realizará las respectivas pruebas de funcionamiento, para determinar el análisis de desempeño del mismo.

3.2 SELECCIÓN DE HARDWARE EN BASE A AHP (PROCESO DE ANÁLISIS JERÁRQUICO)

AHP es un Proceso de Análisis Jerárquico, desarrollado en 1970 por el Dr. Thomas Saaty, que permite la toma de decisiones en base a ponderar prioridades cuando se debe considerar aspectos tanto cuantitativos como cualitativos en una decisión.

Este método a más de presentar un sustento matemático, permite incluir la participación de diversas personas o grupos de interés.

Para el presente proyecto de titulación el grupo de interés para la selección de hardware, son los administradores de red o ingenieros de pequeñas y medianas empresas de la cuidad de Quito, debido a su conocimiento en la adquisición de hardware y en los equipos de conectividad en este caso ruteadores.

Como la población a analizar es extensa se debe seleccionar una muestra que sirva como base para valorar la importancia de cada aspecto en la selección de hardware.

3.2.1 CÁLCULO DE LA MUESTRA

3.2.1.1 Metodología de la Investigación

Se entiende por metodología al establecimiento de teorías sobre el método. Entonces la metodología es la descripción y análisis de los métodos. Por lo que se puede afirmar que la metodología es el estudio analítico y crítico de los métodos de investigación y de prueba, esto incluye: la descripción, el análisis y la valoración crítica de los métodos que conciernen a la investigación[40].

3.2.1.2 Tipo de Investigación

La elección del tipo de investigación determinará los pasos a seguir del estudio, sus técnicas y métodos que puedan emplear en el mismo. En general determina todo el enfoque de la investigación influyendo en instrumentos, y hasta la manera de cómo se analiza los datos recaudados.

Existen dos tipos principales de investigación: de Campo o de Laboratorio, éstas a su vez pueden clasificarse en cuatro tipos principales:

---

[40] Villalba Avilés, Carlos 2006:21

3.2.1.2.1 De Campo

Este tipo de investigación se apoya en informaciones que provienen entre otras, de entrevistas, cuestionarios, encuestas y observaciones. Como es compatible desarrollar este tipo de investigación junto a la investigación de carácter documental, se recomienda que primero se consulten las fuentes de la investigación de carácter documental, a fin de evitar una duplicidad de trabajos.

Se usó esta metodología porque permite el contacto directo con el objeto de estudio, en este caso se pretende crear un nuevo producto totalmente enfocado a Pequeñas y Medianas Empresas ubicadas en la ciudad de Quito.

3.2.1.2.2 Documental o Bibliográfica

Es el proceso ordenado y lógico, de pasos para realizar una investigación documental sobre algún problema que nos inquiete, interese o preocupe, cuyos resultados serán de validez científica. Con la investigación documental se puede elaborar un marco teórico conceptual para formar el cuerpo del objeto de estudio.

El presente proyecto toma en cuenta varios materiales bibliográficos como documentos, libros de bibliotecas, e Internet que han permitido fundamentar con bases teóricas científicas el estudio de todos los aspectos científicos y técnicos que respaldan el trabajo investigado.

3.2.1.2.3 Descriptiva

Las investigaciones descriptivas buscan caracterizar las propiedades importantes de personas, grupos comunidades o cualquier otro elemento – fenómeno que pueda ser sometido a un análisis. Cuando se describe, se ha aprendido las múltiples partes de un objeto de estudio. Esta captación sirve para profundizar el conocimiento objetivo y más tarde elaborar ciertos conceptos, leyes y categorías.

3.2.1.2.4 Explicativa

La investigación explicativa intenta dar cuenta de un aspecto de la realidad, explicando su significatividad dentro de una teoría de referencia, a la luz de leyes o generalizaciones que dan cuenta de hechos o fenómenos que se producen en determinadas condiciones.

3.2.1.3 **Población y Muestra**

A la población se la define como el conjunto de todos los casos que concuerdan con una serie de especificaciones. La realización de este proyecto implica la estimación de una característica de alguna población. Mediante la técnica de la población y muestra se evaluará al grupo de personas relacionadas directamente con el proyecto planteado. A la muestra se la define como el desagregado de la población. Una muestra tiene características que le corresponden a la población.

La siguiente fórmula nos permite determinar el tamaño de la muestra, para obtener datos reales en la Ciudad de Quito a la cual se enfoca el presente proyecto.

$$n = \frac{NZ_c^{\,2} pq}{d^2(N-1) + Z_c^{\,2} pq}$$

Donde:

n: Tamaño de la muestra a obtener.

N: Tamaño de la población.

d: Porcentaje de estimación de error admisible.

Z_c: Coeficiente de confianza.

p: Proporción muestral.

q: 1-p.

A la hora de obtener el tamaño de la muestra, que representen la totalidad de opiniones del estudio, es necesario conocer los datos con los que se cuenta, se

escogió el método denominado: Aleatorio Simple conociendo el tamaño del universo a encuestar.

Para conocer el tamaño del universo se tomó el estudio de PYMES con Base Tecnológica definida (Infraestructura de Redes) realizado por el INEC en el año 2006[41], dando como resultado 1800 empresas a nivel nacional con esta categoría, de las cuales el 30% se encuentran en la ciudad de Quito, que representa un total de 540 empresas.

Se desea tener como máximo un 15% de coeficiente de error, debido a la antigüedad del estudio con el que se realizó la muestra y se desea dar una confianza del 90%.

$$n = \frac{z^2 pqN}{e^2(N-1) + z^2 pq}$$

Donde:

z = grado de confianza

e = error permisible

p = proporción de la muestra a favor

q = proporción de la muestra en contra

N = tamaño de la población

Para ejecutar el cálculo se establece los siguientes valores:

z: El grado de confianza, que es la cantidad de valores válidos, se establece un valor de un 90% dando un valor de 1,645.

e: Error permisible, con el objeto de asegurar que el muestreo se acerque a la realidad se establecen valores altos de error esto es de un 15%.

[41] www.ecuadorencifras.com/cifras-inec/main.html

p y q: Proporción de la muestra a favor, se establece una equidad entre la proporción de la muestra a favor y en contra con un valor del 50% para cada proporción.

N: El tamaño de la población de PYMES de acuerdo al estudio realizado por el INEN en el año del 2006 es 540.

- **Cálculo del Tamaño de la Muestra**

MUESTRA CON PROPORCIONES	
VARIABLE	VALOR
Población	540
Intervalo de confianza	90%
Z	1,645
Probabilidad a favor	50%
Probabilidad en contra	50%
Error de estimación %	15%
Tamaño de la muestra	**29**

Tabla 3-1 Cálculo del tamaño de la muestra

3.2.1.4 Establecimiento de los aspectos a considerar en la selección de hardware

Luego de determinar los actores que deben participar en la toma de decisión, y el tamaño de la muestra se debe definir qué aspectos son importantes para la compra de hardware.

La razón más importante para evaluar a un equipo es el motivo por el que se piensa adquirirlo, en este caso la razón fundamental es la implementación de un ruteador, luego de este análisis los aspectos que consideramos importantes son:

- ***Precio.-*** Uno de los objetivos planteados en el presente proyecto de titulación es ofrecer una solución de bajo costo para pequeñas y medianas empresas, que represente una buena solución en cuanto a funcionalidad y a precio. Además se considera importante mantener equilibrio entre las

necesidades presentes en la compra de un equipo y considerar que el mismo tenga la flexibilidad para evolucionar o crecer.

- ***Características de hardware.-*** Este parámetro se considera para la toma de decisión debido a que en la actualidad existen una gran variedad de computadores con excelentes características en cuanto a procesador, memoria, disco duro, sin embrago resulta un gasto excesivo comprar un equipo con excesivas características en hardware si no se va a aprovechar todo su potencial. Además se deben examinar los periféricos que se conectan a un tipo de computadora, si bien hoy en día la tendencia de los constructores de periféricos es la flexibilidad en las conexiones y a no depender de un solo tipo de estándar que los pudiera poner fuera del mercado, se debe analizar cuidadosamente estas características para no tener gastos innecesarios.
- ***Soporte y Repuestos.-*** Cuando se adquiere un equipo no solo se debe considerar sus características de hardware y la disponibilidad en el mercado de las mismas, sino además se debe investigar si existe soporte técnico y los respectivos repuestos ante cualquier daño en el hardware, ya que sería un gasto adicional si una pieza llegara a dañarse en el prototipo y no pueda ser adquirida dentro del país.
- ***Disponibilidad en el mercado y garantía.-*** Se debe analizar si todo el hardware a ser adquirido esté disponible en el mercado a fin de evitar costos de importación, otro parámetro a considerar es la garantía que nos proporcione el distribuidor del equipo.
- ***Marca.-*** Existe una gran variedad de marcas y modelos en el mercado, si bien los estándares y la imagen corporativa de los fabricantes de hardware es importante en la toma de decisión, se debe investigar si dichas marcas cumplen con los estándares básicos que certifiquen un funcionamiento adecuado.
- ***Recomendación del fabricante de software.-*** Es importante tener claro las aplicaciones que van a correr, su tamaño, los paquetes de información, para la selección de una determinada compra de hardware, debido a que

ciertos sistemas operativos y programas requieren más capacidad en cuanto a memoria y almacenamiento.

3.2.2 AHP - PROCESO DE ANÁLISIS JERÁRQUICO - ANALYTIC HIERARCHY PROCESS

Previo a la aplicación de este método, se analizó la población a encuestar, la muestra de la cual se obtendrán los resultados y se identificó los criterios para la evaluación y priorización en la selección de hardware del prototipo.

Como primer paso es elaborar una matriz de comparación pareada en la cual se ubican en la primera fila y columna los aspectos a consideran en la selección de hardware, en la tabla 3.2 se indica la ubicación de cada parámetro. C se refiere a Columna y F a Fila.

3.2.2.1 Matriz de Comparación Pareada

El AHP permite comparar parejas de opciones (por ejemplo, Precio vs. Características de hardware, Disponibilidad en el mercado vs. Marca, etc.)

Las celdas dónde se cruzan elementos idénticos (Ejemplo: Precio vs. Precio) se coloca un valor de 1. Con esto la Matriz de Comparación Pareada queda inicializada.

AHP	Precio (C1)	Características de hardware (C2)	Soporte y Repuestos (C3)	Disponibilidad en el mercado y garantía(C4)	Marca (C5)	Recomendación del fabricante de software(C6)
Precio (F1)						
Características de hardware (F2)						
Soporte y Repuestos (F3)						
Disponibilidad en el mercado y garantía (F4)						
Marca (F5)						
Recomendación del fabricante de software (F6)						

Tabla 3-2 Matriz de comparación pareada

3.2.2.2 Matriz de Comparación Pareada Inicializada

Para la inicialización de la matriz se coloca 1 en las celdas donde se cruzan los elementos idénticos, en la Tabla 3.3 se muestra un ejemplo.

AHP	Precio (C1)	Características de hardware (C2)	Soporte y Repuestos (C3)	Disponibilidad en el mercado y garantía (C4)	Marca (C5)	Recomendación del fabricante de software (C6)
Precio (F1)	1					
Características de hardware (F2)		1				
Soporte y Repuestos (F3)			1			
Disponibilidad en el mercado y garantía (F4)				1		
Marca (F5)					1	
Recomendación del fabricante de software (F6)						1

Tabla 3-3 Matriz de comparación inicializada

- Comparación 1: Precio (C_1) vs. Precio (F_1).

Esta comparación es trivial (son el mismo elemento), el valor se inicializa como 1.

- Comparación 2: Precio (C_1) vs. Características de hardware (F_2).

En esta comparación, se debe decidir qué elemento es más importante. Para facilitar esta decisión, se debe usar la Tabla de Ponderación que se presenta a continuación. Esta tabla se debe usar para cualquier ejercicio de AHP, no sólo para este caso particular.

Tabla de Ponderación

ESCALA NUMÉRICA	ESCALA VERBAL	EXPLICACIÓN
1.0	Ambos elementos son de igual importancia.	Ambos elementos contribuyen con la propiedad en igual forma.
3.0	Moderada importancia de un elemento sobre otro.	La experiencia y el juicio favorece a un elemento por sobre el otro.
5.0	Fuerte importancia de un elemento sobre otro.	Un elemento es fuertemente favorecido.

7.0	Muy fuerte importancia de un elemento sobre otro.	Un elemento es muy fuertemente dominante.
9.0	Extrema importancia de un elemento sobre otro.	Un elemento es favorecido, por lo menos con un orden de magnitud de diferencia.
2.0,4.0,6.0,8.0	Valores intermedios entre dos juicios adyacentes.	Usados como valores de consenso entre dos juicios.
Incrementos de 0.1	**Valores intermedios en la graduación más fina de 0.1 (Por ejemplo 5.2 es una entrada válida).**	**Usados para graduaciones más finas de los juicios.**

Fuente: http://www.rlc.fao.org/proyecto/139jpn/document/3dctos/sirtplan/infotec/2_AHP.pdf

Tabla 3-4 Tabla de ponderación

En el **ANEXO B**, se da un ejemplo de cómo se realiza la comparación de importancia entre cada parámetro, los respectivos cálculos y finalmente como se analizan los resultados obtenidos.

Además se incluye las encuestas realizadas a cada participante en esta toma de decisión.

3.2.3 ANÁLISIS DE RESULTADOS

Luego de las encuestas(*) realizadas a una muestra de 29 personas encargadas de la adquisición de infraestructura de redes de las distintas pequeñas y medianas empresas de la ciudad de Quito, se obtuvieron los siguientes resultados:

(*) Para mayor detalle acerca de las encuestas, remítase al ANEXO B

Parámetros	Porcentaje	Prioridad
Recomendación del fabricante de software.	33,67%	1
Características de hardware.	19,44%	2
Precio.	14,97%	3
Disponibilidad en el mercado y garantía.	11,05%	4
Soporte y Repuestos.	11,02%	5
Marca.	9,85%	6

Tabla 3-5 Matriz de resultados

En base a las prioridades obtenidas, se realizará el proceso de selección, considerando que la recomendación del fabricante de software fue el parámetro de mayor importancia.

A continuación se realiza un estudio de los cinco ítems con mayor prioridad.

3.2.3.1 Recomendación del fabricante de software

En la Tabla 3.6 se muestra las recomendaciones dadas por el fabricante del sistema operativo, del fabricante de software de enrutamiento, del fabricante de software de IPsec y de QoS.

Recomendaciones dadas por el fabricante del Sistema Operativo, Linux Debian.	
Arquitecturas:	I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC, ALPHA, MIPS, MIPSEL, SOURCE, MULTI-ARCH, ARM.
Requisitos de hardware mínimos:	32 MB de RAM.
	4 GB de Disco Duro.
Recomendaciones dadas por el fabricante de Software de Enrutamiento XORP.	
Hardware:	Lo necesario para soportar un sistema operativo Linux con kernel 2.6.
	4 GB de Disco Duro.
Recomendaciones dadas por el fabricante de Software del estándar de seguridad para IP.	
Hardware:	Lo necesario para soportar un sistema operativo Linux con kernel 2.6.
	0.5 GB de Espacio en Disco Duro.
Recomendaciones dadas por el fabricante de Software Calidad de Servicio QoS.	
Hardware:	Lo necesario para soportar un sistema operativo Linux con kernel 2.6, con módulos de Calidad de Servicio Habilitado.
	300 KB de Espacio en Disco Duro.

Tabla 3-6 Recomendación del fabricante de software

Por lo que se obtiene:

Arquitectura: I386, IA64, AMD64, SPARC, HPPA, S390, POWERPC, ALPHA, MIPS, MIPSEL, SOURCE, MULTI-ARCH, ARM.
Memoria RAM: 160 MB.
Disco Duro: 9 GB.

Tabla 3-7 Resultados de recomendación del fabricante de software.

3.2.3.2 Características de hardware

3.2.3.2.1 Dimensionamiento de Procesador

La frecuencia del procesador es el parámetro más importante de los componentes de hardware, ya que al tratarse de una solución de enrutamiento basado en software el componente de hardware que realizara el manejo de los paquetes será el procesador.

Consideraciones:

- El enrutador contará con 3 tarjetas de red con capacidad de 1 Gbps, por lo que durante periodos de carga máxima podría manejar hasta 3 Gbps.
- Se transmitirán tramas IP de 1500 Bytes, lo que representa 12000 bits.
- El consumo máximo de procesador para operar durante periodos de carga máxima no superará el 75%.

$$Frecuencia[MHz] = \frac{\#Interfaces\ x\ Trafico\_Maximo[Mbps]}{Tramas[b]}$$

$$Frecuencia[MHz] = \frac{3x1000[Mbps]}{12000[b]}$$

$$Frecuencia\,[MHz] = \frac{3000M}{12000}\left[\frac{1}{s}\right]$$

$$Frecuencia = 0.25\,[MHz]$$

3.2.3.2.2 Dimensionamiento de Disco Duro

El disco duro es un componente fundamental y crítico para el funcionamiento de enrutador ya que aquí se alojan los diferentes componentes de hardware, aunque las recomendaciones de los fabricantes del software no son significativas con relación a las capacidades ofrecidas en el mercado este parámetro es muy importante en el dimensionamiento del equipo.

Consideraciones:

Software	*Espacio en disco recomendada por el fabricante de software*
Sistema Operativo GNU LINUX DEBIAN.	4 GB
Enrutamiento XORP.	4 GB
Seguridad IP RACOON.	0.5 GB
Calidad de Servicio CBQ.INIT.	0.01GB
Log del sistema.	5 GB
TOTAL 1.	**13,51 GB**
Crecimiento futuro y respaldo (30 %)	4.06 GB
TOTAL 2.	**17,57 GB**

Tabla 3-8 Cálculo de capacidad de disco duro

3.2.3.2.3 Dimensionamiento de la Memoria

El dimensionamiento de la memoria lo obtendremos de los requisitos del fabricante de software para cada software utilizado.

Software	Requerimiento de memoria
DEBIAN	32 MB
XORP	128 MB
CBQ-INIT	128 MB
TOTAL	288 MB

Tabla 3-9 Dimensionamiento de Memoria.

La capacidad de la memoria se mide en múltiplos de byte (8 bits): kilobytes (1.024 bytes) y megabytes (1.024 kilobytes). Los valores definidos son 256 MB, 512 MB, 1 GB, 2 GB, etc. Para el prototipo hemos considerado 1 memoria de 1GB, debido a que el valor entre la memoria de 512MB y la de 1GB no es considerable.

En base a las características del Hardware se tiene que las partes, deben poseer las siguientes características:

Procesador	Arquitectura: 64 bits. Procesador: Doble Núcleo. Frecuencia: 2.1 GHz mínimo.
Memoria	1 dims 1GB. Frecuencia 800 MHz.
MotherBoard	Frecuencia de memorias, mínimo 2 slots de 800 MHz. Bus de sistema: mínimo 800 MHz. Puertos de Expansión PCI = 3. Interfaces SATA = 2. Interfaces IDE = 1.
Disco Duro	Cantidad: 1. Revoluciones: mínimo 7200 rpm. Tamaño: mínimo 80 GB.
Interfaces de Red	Cantidad: 3. Velocidad: 10/100 Mbps

Tabla 3-10 Características de hardware

3.2.3.3 Disponibilidad en el mercado y garantía

En base a la disponibilidad en el mercado se encontró las siguientes marcas como las más representativas del mercado local y cuentan con canales de distribución que permiten aplicar las garantías.

Procesador	Memoria	Tarjeta Madre	Disco Duro	Interfaces de Red
Intel	Kingston	Intel	Samsung	3com
AMD	Markvisión	Mis	Maxtor	Dlink
	Adata	Biostar	Sagate	Cnet
	Corsair			

Tabla 3-11 Disponibilidad en el mercado y garantía

3.2.3.4 Soporte y Repuesto

En base a soporte y repuestos los siguientes componentes cuentan con soporte localmente en el país y garantizan repuestos de los equipos.

Procesador	Memoria	Motherboard	Disco Duro	Interfaces de Red
Intel	Kingston	Intel	Samsung	3com
AMD	Adata	Mis	Maxtor	Dlink

Tabla 3-12 Soporte y repuestos

3.2.3.5 Precio

En base a los equipos disponibles en el mercado, se seleccionará tomando como referencia el precio más bajo.

Procesador		**Precio (USD)**
Intel	INTEL DUAL CORE E2200 2.2 Ghz.	**77,00**
AMD	AMD ATHLON AMD2 5400 2.8 DC X2.	**70,00**
Memoria		
Kingston	KINGSTON 1GB PC-667.	**13,00**
Adata	Adata1GB PC-800.	**10,00**
Tarjeta Madre		
Intel	INTEL DG31PR S775,1333 Ghz,DDR2,V,S,R.	**66,00**
Msi	MSI AMD2 K9N2G PHENON, V,S,R.	**89,00**
Disco Duro		

Samsung	160 GB SAMSUNG SATA 7200 RPM.	**39,00**
Maxtor	160 GB MAXTOR IDE 7200 RPM.	**42,00**
Interfaces de Red		
3com		
Dlink	Giga bit Ethernet D-LINK modelo DGE-530T	**30,00**

Tabla 3-13 Tabla comparativa de precios

Después del proceso de selección se obtuvo que las características de hardware para el enrutador Linux sean las siguientes:

Ítem	Cantidad	Característica	Precio Unitario (USD)	Precio Total (USD)
Procesador	1	**INTEL DUAL CORE E2200 2.2 Ghz.**	77	77
Memoria	2	**ADATA 1GB PC-800.**	10	20
Tarjeta Madre	1	**INTEL DG31PR S775,1333 Ghz,DDR2,V,S,R.**	66	66
Disco Duro	1	**160GB SAMSUNG SATA 7200 RPM.**	42	42
Red	2	**Giga bit Ethernet D-LINK modelo DGE-530T.**	30	60
Case	1	**CASE DLUXE DLC-MF453 MIDTOWER 24P.**	39	39
TOTAL				**304**

Tabla 3-14 Tabla de partes y precios

3.3 CONFIGURACIÓN E IMPLEMENTACIÓN DEL PROTOTIPO

Como primer paso se debe instalar el Sistema Operativo para este proceso se debe verificar que el hardware cumpla con los requisitos mínimos establecidos en el proceso de selección y dimensionamiento; para asegurar la compatibilidad entre los componentes y el adecuado funcionamiento del software. En el **ANEXO C**, se detalla el proceso de instalación del sistema operativo Linux Debian.

3.3.1 RECONFIGURACIÓN DEL KERNEL

Los objetivos de realizar una recopilación del kernel son:

- Obtener un sistema más rápido, estable y robusto.
- Obtener un sistema con soporte a elementos de hardware no encontrado en kernels anteriores.
- Obtener un sistema con soporte a características especiales disponibles pero no habilitadas en kernels anteriores.

El proceso de personalizar el kernel nos permite sacar un mayor provecho de las diferentes características que ofrece el software.

Particularmente para el proyecto Router Linux, se busca eliminar los módulos innecesarios y activar los módulos de Calidad de Servicio e IPsec que están soportados nativamente pero no vienen activos por defecto.

Es importante recordar que no se debe utilizar kernels de desarrollo en equipos de producción, ya que no se garantiza la estabilidad del sistema operativo.

En el **ANEXO D** se muestra el proceso de reconfiguración del kernel.

3.3.1.1 **Tiempo de compilación**

El tiempo necesario para compilar es muy variable, dependiendo del equipo que se vaya a utilizar y la cantidad de módulos que desee agregar o eliminar de la configuración por defecto. En equipos recientes, el proceso toma cerca de 10 minutos o menos. Se debe tener en cuenta que si utiliza computadores poco recientes el proceso puede tardar incluso varias horas.

3.3.1.2 **Espacio de disco requerido**

El espacio necesario para compilar el kernel varía de acuerdo a la versión del kernel a utilizar. Las fuentes del kernel 2.6.26 ocupan cerca de 42 MB, así que para este caso tomado de ejemplo sería bueno disponer de al menos unos 50 MB libres.

3.3.1.3 **Descripción de las características del kernel**

3.3.1.3.1 Processor Type and Features - Tipo de procesador y características

Bajo esta sección es posible definir entre otros:

- Tipo de procesador del equipo.
- Emulación del procesador matemático (útil con equipos 386 y similares que vienen sin él). Deshabilitada en forma predeterminada.
- Soporte para sistemas con más de 1 procesador (SMP). Se deshabilita este soporte si va a trabajar en un computador que solamente dispone de 1 procesador.

Es posible que el kernel no arranque si se selecciona un procesador incorrecto,

Para el ejemplo, el cambio principal realizado fue habilitar el soporte para SMP.

3.3.1.3.2 Plug and Play

Se habilita esta opción si se desea que el kernel automáticamente configure algunos dispositivos periféricos como memorias USB.

3.3.1.3.3 Block Devices – Dispositivos de Bloque

Bajo esta sección se presenta soporte para:

- Unidades de discos flexibles.
- Soporte mejorado para dispositivos de almacenamiento estándar IDE y otros dispositivos de datos en modo bloques.

Si se dispone de un disco duro IDE, será necesario habilitar el soporte para el disco IDE apropiado.

Para el ejemplo, se aceptaron los valores predeterminados propuestos sin realizar ningún cambio.

3.3.1.3.4 Networking Options – Opciones para Redes

Esta sección es de vital importancia para el funcionamiento del equipo. Las opciones que en forma predeterminada vienen habilitadas son:

- Packet socket.
- Unix domain socket.
- TCP/IP networking.
- IP: Allow large Windows.

Se habilitaron todos los módulos contenidos sobre estas opciones que permiten a GNU/Linux actuar como un enrutador y dar soporte a protocolos de TCP/IP, Calidad de Servicio e IPsec.

3.3.1.3.5 Telephony Support – Soporte a Telefonía

Esta opción permite dar soporte a correo de voz, fax y otros dispositivos multimedia que interoperan con módems; viene deshabilitada en forma predeterminada.

3.3.1.3.6 Network Device Support – Soporte a Dispositivos de Red

Entre las diferentes alternativas disponibles en esta sección se destacan:

- Ethernet 10/100/1000.
- PPP (point-to-point) support.

Respecto a la primera opción, al seleccionarla aparece un listado de tarjetas de red disponibles, incluyendo entre otros, tarjetas 3COM, tarjetas EISA, VESA, PCI y controladores integrados en las tarjetas principales, y otras tarjetas ISA. Por otro lado, el soporte para protocolo PPP es necesario para permitir conexión a Internet por vía telefónica.

Para el ejemplo, se habilitó soporte integrado al kernel para la tarjeta de red disponible y en forma modular para otras tarjetas de red de las cuales

eventualmente se pueda disponer. En este complemento, se habilitó el soporte para protocolo PPP. Desactivado en forma predeterminada.

3.3.1.3.7 USB Support – Soporte USB

Se habilita esta opción si el equipo tiene puerto USB.

3.3.1.3.8 File Systems – Sistema de Archivos

Esta sección determina el tipo de sistemas de archivos a utilizar. Entre las opciones habilitadas en forma predeterminada se destacan:

- ISO 9960 CDROM filesystem support, para poder acceder a CDROM estándares.
- /proc filesystem, sistema de archivos virtual que proporciona información sobre el sistema operativo.
- Second extended fs support, el sistema de archivos estándar de GNU/Linux.
- NFS filesystem support, para interactuar con otros equipos UNIX utilizando el protocolo NFS.

3.3.1.3.9 Console Drivers – Controladores de Consola

Bajo esta sección se presentan dos opciones:

- VGA text mode.
- Video mode seleccion support.

La primera de ellas, habilitada en forma predeterminada, permite utilizar GNU/Linux en modo texto en una pantalla que cumpla con el estándar VGA genérico. La segunda opción, deshabilitada en forma predeterminada, permite aprovechar algunas altas resoluciones en modo texto que podría ofrecer el BIOS, lo que para el caso particular del enrutador se mantiene desactivado.

3.3.1.3.10 Kernel Hacking

Permite tener cierto control sobre el sistema aún si el sistema se cae haciendo uso de la tecla de Peticiones del Sistema (SysRQ).

3.3.2 INSTALACIÓN Y CONFIGURACIÓN DE SOFTWARE DE ENRUTAMIENTO XORP

XORP es un paquete de software escrito en C + +. Para compilar XORP se requiere cerca de 1,4 GB de espacio libre en disco, y hasta una hora dependiendo de la velocidad de la CPU y la versión del compilador que se está usando. Para compilar XORP se debe tener gmake instalado.

Se compila XORP escribiendo dentro del directorio /etc/xorp:

```
# ./configure
# gmake
```

XORP está constituido por múltiples procesos, dependiendo de los protocolos utilizados de cómo se muestra en la figura a continuación:

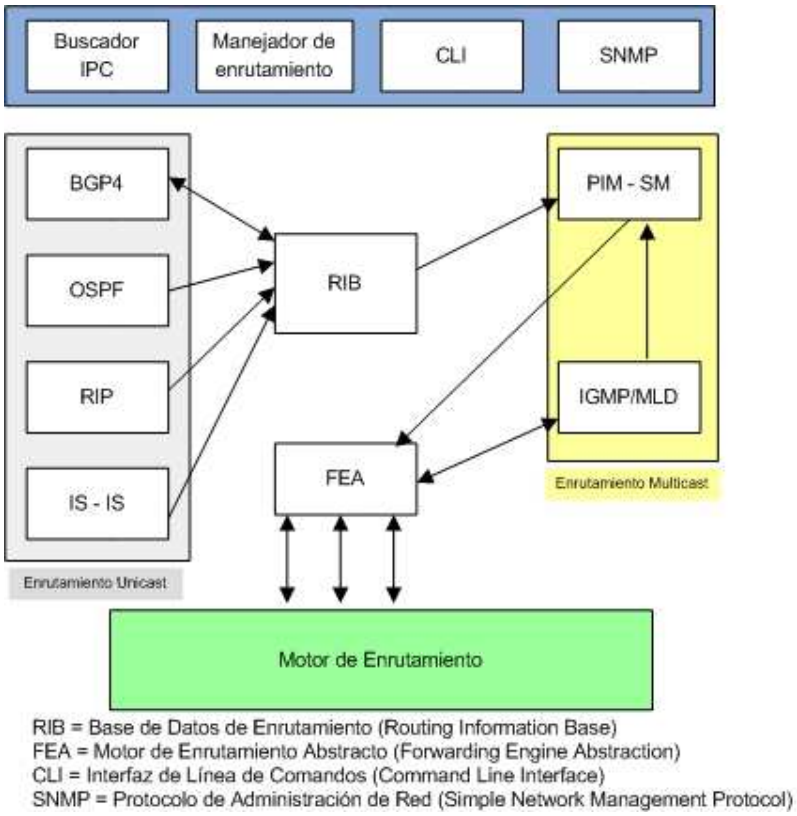

Figura 3- 1 Arquitectura XORP

Existe un proceso que gestiona el enrutador XORP, este se llama xorp_rtrmgr (XORP router manager). Normalmente xorp_rtrmgr debe ejecutarse como root. Esto se debe a que pone en marcha procesos que requieren un acceso privilegiado.

XORP contiene un archivo de configuración en el mismo directorio por defecto, este archivo de configuración que se llama config.boot, aquí se aplican las direcciones IP, interfaces y protocolos de enrutamiento para que coincida con el entorno de red local.

Un enrutador con XORP debe estar configurado para realizar las operaciones que desee, la información de configuración se puede proporcionar en una de las dos maneras:

- Usando un archivo de configuración cuando se inicia la rtrmgr. Por defecto, el rtrmgr carga la configuración desde el archivo "config.boot". Este archivo

puede ser especificado por la opción "-b <filename>" opción de línea de comandos:

```
xorp_rtrmgr -b my_config.boot
```

- Utilizando xorpsh en la interfaz de línea de comandos después que inicie rtrmgr. El xorpsh tiene que estar en modo de configuración:

```
user@hostname> configure
    Entering configuration mode.
    There are no other users in configuration mode.
    [edit]
    user@hostname#
```

3.3.2.1 **Configuración de XORP**

Para interactuar con un enrutador XORP utilizando la interfaz de línea de comandos se ejecuta el comando "xorpsh". Esto permite la configuración del enrutador y el seguimiento del estado del mismo.

El comando xorpsh proporciona una consola de comandos interactiva con el usuario, xorpsh podría ser establecido como un usuario de la consola que da acceso al enrutador a través del protocolo ssh.

```
user@hostname>
```

Ejecutamos control-d, para salir en cualquier momento de xorpsh. Al escribir "?" nos muestra una lista de los comandos disponibles.

3.3.2.2 **Modos de operación de XORP**

Xorpsh tiene dos modos de comando:

- *Modo de operación.-* Permite la interacción con el enrutador para supervisar su funcionamiento y el estado.

- *Modo de configuración.-* Permite al usuario ver la configuración del enrutador, cambiarla, cargar y guardar configuraciones de archivo.

En términos generales, el modo operativo no tiene acceso privilegiado; un usuario no puede escribir algo que impacte o altere el funcionamiento del enrutador. En cambio, el modo de configuración permite que todos los aspectos de funcionamiento del enrutador puedan ser modificados.

3.3.2.3 **Comandos**

Modo de Operación

```
user@hostname> ?
Possible completions:
configure       Switch to configuration mode
exit            Exit this command session
help            Provide help with commands
quit            Quit this command session
show            Display information about the system
```

configure: cambia el modo de funcionamiento a modo de CONFIGURACIÓN

exit: salida de xorpsh.

help: proporciona ayuda en línea.

quit: Cierra xorpsh. Es equivalente a la salida de comando exit.

show: muestra muchos aspectos de la gestión y configuración del equipo.

3.3.2.4 **Modo de Configuración**

En el modo de configuración, el símbolo del cambio de usuario de user@hostname> pasa a user@hostname#.

```
user@hostname> configure
Entering configuration mode.
There are no other users in configuration mode.
[edit]
user@hostname#
```

En el modo de configuración al digitar "?" nos presenta una lista más amplia de comandos que el modo de operación.

```
[edit]
user@hostname# ?
Possible completions:
Commit           Commit the current set of changes
Create           Alias for the ''set'' command (obsoleted)
Delete           Delete a configuration element
Edit             Edit a sub-element
Exit             Exit from this configuration level
Help             Provide help with commands
Load             Load configuration from a file
Quit             Quit from this level
Run              Run an operational-mode command
Save             Save configuration to a file
Set              Set the value of a parameter or create a new
                 element
Show             Show the configuration (default values may be
                 suppressed)
Top              Exit to top level of configuration
Up               Exit one level of configuration
user@hostname#
```

3.3.2.5 **Interfaces de red**

Cada dispositivo de red físico en el sistema se considera una "interfaz". Cada interfaz puede contener una serie de interfaces virtuales ("vif"). En la mayoría de los casos el nombre de la interfaz y vif son idénticos. Una interfaz virtual está configurada con la dirección o direcciones que serán utilizadas.

En la configuración de cada interfaz se debe colocar la "address" (dirección IP), "prefix-length" (longitud del prefijo) y la dirección de "broadcast". Con los archivos de configuración de cada router se explicará el detalle de cada comando utilizado.

3.3.2.6 **FEA Forwarding Engine Abstraction – Motor de Enrutamiento Abstracto**

Se trata de un requisito para permitir el reenvío de cada familia de protocolos. Con esta configuración el enrutador recibe y envía paquetes de una interfaz a otra.

Se puede configurar esta opción en un enrutador para que por ejemplo sólo transmita paquetes IPv6 o solo paquetes IPv4.

Con la siguiente configuración se permite al enrutador el reenvío de IPv4. En el archivo de configuración de cada router se explicará en detalle la sintaxis del código.

```
fea {
    unicast-forwarding4 {
      disable: false
      forwarding-entries {
          retain-on-startup: false
          retain-on-shutdown: false
      } }
```

Si el sistema soporta IPv6 y se requiere el reenvío de IPv6, en la declaración inicial se coloca unicast-forwarding6.

3.3.2.7 **DIRECCIONAMIENTO IP VERSIÓN 4**

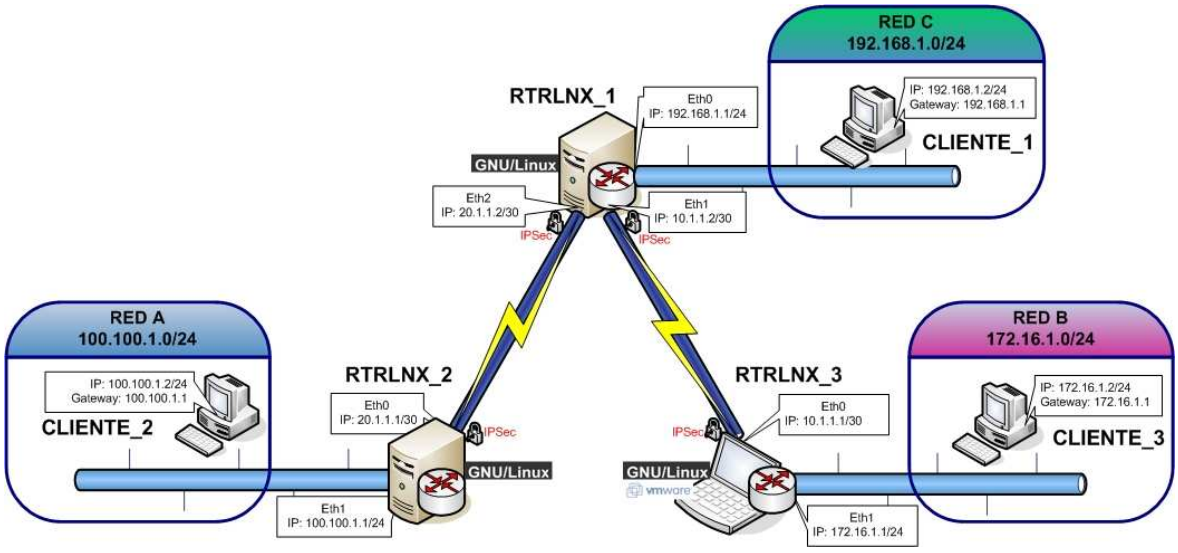

Figura 3- 2 Direccionamiento IPv4

En la figura 3.2 se indica, el diagrama de red elaborado para probar el funcionamiento del prototipo implementado, la configuración de las interfaces de red de cada enrutador se encuentra en **ANEXO E.**

A continuación se detallará la configuración del enrutador de dirección IP: 192.168.1.1.

ENRUTADOR RED C	
Dirección de Interface 0:	192.168.1.1
Sub Máscara:	255.255.255.0
Dirección de Interface 1:	10.1.1.2
Sub Máscara:	255.255.255.252
Dirección de Interface 2:	20.1.1.2
Sub Máscara:	255.255.255.252

Tabla 3-15 Configuración enrutador RTRLNX_1

3.3.2.7.1 Configuración RTRLNX_1

En el archivo de configuración se coloca el código para cada interfaz de red, se coloca la dirección de red, la longitud del prefijo y la dirección de broadcast.

Este enrutador tiene las siguientes interfaces:

- La ethernet 0 para conectarse a la red 192.168.1.0, esta es la red local.
- La ethernet 1 para conectarse a la red 172.16.1.0.
- La Ethernet2 para conectarse a la red 100.100.1.0.

Los parámetros de configuración se utilizan de la siguiente manera:

interfaces: esta declaración delimita toda la información de configuración de la interfaz, dentro del archivo de configuración de XORP.

restore-original-config-on-shutdown - restaurar-configuración-original-en-shutdown: este indicador permite el restablecimiento de la configuración de la red original cuando se cierra FEA, es decir cuando ya no se permite el reenvío de paquetes de una interfaz a otra.

Permite un valor booleano, si es asignado true, entonces el restablecimiento se habilita. El valor por defecto es falso (es decir, no restablecer la configuración de red).

interface - Interfaz: delimita la configuración de una interfaz. Para cada interfaz, la siguiente configuración es posible:

- *description - descripción:* se utiliza principalmente para ayudar a recordar el propósito de la interfaz. Es opcional.
- *default-system-config - configuración por defecto:* normalmente todas las interfaces, vifs, y las direcciones de un router XORP se configura a través del archivo de configuración XORP. Sin embargo, en determinadas circunstancias, es útil ejecutar XORP como un demonio de enrutamiento sin cambiar la configuración actual de las interfaces y las direcciones. Si por defecto-system-config se utiliza, entonces las interfaces virtuales vif, direcciones ip no debe ser configuradas. Es por ello que se comenta esta línea de código ya que configuramos cada interfaz de red.
- *disable - desactivar*: este indicador desactiva o activa la interfaz para el enrutamiento y transmisión. Toma valores de verdadero o falso. Configurando una interfaz con disable= true, elimina su configuración

vif - Interfaz virtual: Permite la configuración de interfaces virtuales dentro de una interfaz física. Para cada vif, la configuración posible es:

- *disable - desactivar*: este indicador desactiva o activa la vif para el enrutamiento y transmisión.
- *address - dirección*: especifica una nueva dirección IP para la interfaz virtual vif. Una sola vif puede tener múltiples direcciones IP, además puede configurarse en una misma vif direcciones IPv4 e IPv6. Para cada dirección, las siguientes opciones son posibles:
    - *prefix-leng - longitud del prefijo:* determina la longitud de prefijo de la subred conectada a esta interfaz. Este campo es obligatorio para cada una de las direcciones.
    - *broadcast:* Se configura la dirección de broadcast para cada interfaz virtual vif. Sólo es necesario para las direcciones IPv4. Por ejemplo para la dirección 192.168.1.1 y de longitud de prefijo

24, para determinar la dirección de broadcast se coloca los últimos 8 bits con un valor de uno, por tanto la dirección es 192.168.1.255.

- *disable - desactivar:* este indicador desactiva o activa la dirección IP en la interfaz virtual. Toma el valor de verdadero o falso.

Configuración de Interfaces en RTRLNX_1

```
#Enrutamiento Estático IPv4 RTRLNX Red 192.168
interfaces {                /*Inicio de configuracion de las interfaces
*/
    restore-original-config-on-shutdown: false  /*No reestablece la
configuración de red*/
/*******************************/
/*Configuracion de la Ethernet 0*/
/*******************************/
    interface eth0 {    /*Configuracion de la Interfaz Ethernet0*/
      description: "Local"
      disable: false    /*Permite que no se elimine la configuración*/
      /*default-system-config*/
        vif eth0 {      /*Configuración de la interfaz virtual para la
ethernet0*/
            disable: false  /*Permite que no se elimine la configuración
de la interfaz virtual*/
            address 192.168.1.1 { /*Configuracion de la direccion IP*/
                  prefix-length: 24 /*Asignacion de la longitud del
prefijo*/
                  broadcast: 192.168.1.255 /*Configuracion de la
direccion de broadcast*/
                  disable: false /*Permite que no se elimine la
configuración de la dirección IP*/
                  } } }
/*******************************/
/*Configuracion de la Ethernet 1*/
/*******************************/
    interface eth1 {
      description: "Externa #1"
      disable: false    /*Permite que no se elimine la configuración*/
      /*default-system-config*/
        vif eth1 { /*Configuración de la interfaz virtual para la
ethernet1*/
               disable: false  /*Permite que no se elimine la
configuración de la interfaz virtual*/
            address 10.1.1.2 { /*Configuracion de la direccion IP para la
eth1*/
                  prefix-length: 24 /*Asignacion de la longitud del
prefijo*/
                  broadcast: 10.1.1.255 /*Configuracion de la direccion
de broadcast*/
                  disable: false /*Permite que no se elimine la
configuración de la dirección IP*/
```

```
                  } } }
/*******************************/
/*Configuracion de la Ethernet 2*/
/*******************************/
    interface eth2 {
      description: "Externa #2"
      disable: false /*Permite que no se elimine la configuración*/
        vif eth2 {
                disable: false /*Permite que no se elimine la
configuración de la interfaz virtual*/
           address 20.1.1.2 { /*Configuracion de la direccion IP para la
eth2*/
                  prefix-length: 24 /*Asignacion de la longitud del
prefijo*/
                  broadcast: 20.1.1.255 /*Configuracion de la direccion
de broadcast*/
                  disable: false /*Permite que no se elimine la
configuración de la dirección IP*/

                  } } } }
```

Configuración FEA

FEA (Forwarding Engine Abstraction) - Abstracción en el proceso de Transmisión: delimita la configuración de reenvío de paquetes.

- *unicast-forwarding4 - transmisión unicast IPv4:* esta directiva se utiliza para configurar el reenvío de IPv4.
- *disable - desactivar:* toma el valor verdadero o falso, desactiva o activa el reenvío de todos los paquetes IPv4 en el router. El valor por defecto es falso.
- *forwarding-entries - transmisión de las entradas:* esta directiva se utiliza para configurar las propiedades de la transmisión de IPv4.
  - retain-on-startup - mantener en startup: toma el valor verdadero o falso, y se utiliza para controlar las entradas en la transmisión de paquetes IPv4 al inicio del reenvío de paquetes. El valor por defecto es falso.
  - retain-on-shutdown – mantener en shutdown : toma el valor verdadero o falso, y se utiliza para controlar si el reenvío de paquetes fue deshabilitado. El valor por defecto es falso.

3.3.2.7.1.1 Rutas estáticas

Este es el más simple de los protocolos de enrutamiento en XORP. Permite la instalación de rutas unicast o multicast (ya sea IPv4 o IPv6).

En el archivo de configuración se detalla dos rutas estáticas creadas.

- Ruta a la red 172.16.1.0, como siguiente salto se configura la eth0 de dirección IP 10.1.1.1 del router B.
- Ruta a la red 100.100.1.0, como siguiente salto se configura la eth0 de dirección IP 20.1.1.1 del router C.

Los parámetros de configuración se utilizan de la siguiente manera:

protocols - protocolos: delimita la configuración para todos los protocolos de enrutamiento en la configuración del router XORP.

static - estática: delimita la parte de configuración del router que está relacionada con la configuración de rutas estáticas.

route - ruta: permite especificar la ruta unicast a la que se dirigen los paquetes. Se coloca la dirección IPv4 de la subred destino con su longitud del prefijo. Cada ruta tiene los siguientes atributos:

- *next-hop - próximo-salto:* especifica la dirección IPv4 o IPv6 destino del router de borde de la subred destino.
- *Metric - métricas:* especifica el coste de esta ruta.

Configuración de Rutas estáticas RTRLNX_1

```
#Enrutamiento Estático IPv4 RTRLNX Red 192.168
/*****************************************/
/*Configuracion para el reenvio de paquetes*/
/*****************************************/
fea {
    unicast-forwarding4 { /*habilita el reenvio para paquetes IPv4*/
      disable: false /*Permite que no se elimine la configuración*/
      forwarding-entries { /*se deshabilita las entradas para el
reenvio*/
          retain-on-startup: false
          retain-on-shutdown: false
      } } }
/*Configuracion de rutas estaticas*/
```

```
protocols {
static { /*Configuracion de rutas estaticas*/
route 172.16.1.0/24 { /*Dirección de subred a ser alcanzada*/
      next-hop: 10.1.1.1 /*Dirección de la eth0 del router B*/
        metric: 1 /*Definición de un salto realizado por el paquete para
alcanzar la subred destino*/
            }
route 100.100.1.0/24 { /*Dirección de subred a ser alcanzada*/
      next-hop: 20.1.1.1 /*Dirección de la eth0 del router A*/
        metric: 1/*Definición de un salto realizado por el paquete para
alcanzar la subred destino*/
            } } }
```

3.3.2.7.2 Protocolo De Enrutamiento Dinámico

3.3.2.7.2.1 RIP.

Para ejecutar RIP se especifica el conjunto de interfaces, y las direcciones de las vifs (interfaces virtuales). La configuración de las interfaces es la misma para todos los protocolos de enrutamiento.

Los parámetros de configuración se utilizan de la siguiente manera:

protocols - protocolos: delimita la configuración para todos los protocolos de enrutamiento en la configuración del router XORP.

rip: delimita la configuración del protocolo RIP en el archivo de configuración del router XORP.

export - exportación: la directiva de exportación es una declaración de política. ***interface - interfaz:*** especifica una interfaz de red utilizada por RIP para el enrutamiento. Cada interfaz debe estar configurada en la sección de interfaces del enrutador. Cada interfaz puede tener múltiples vifs configuradas.

vif: especifica una interfaz virtual que debe ser utilizado por RIP para el enrutamiento.

address - dirección: especifica una dirección IPv4 de las interfaces Ethernet que usa RIP para el enrutamiento. Los parámetros que pueden ser especificadas para cada una de las direcciones son:

- *metric - métricas*: especifica la métrica o los costos asociados con las rutas recibidas sobre la dirección de la vif. Antes de decidir la mejor ruta se

añade el coste de la ruta si esta es definida. La métricas debe ser un entero entre 1 y 14, desde 15 en adelante RIP lo consideran infinito.

- *disable - desactivar:* toma el valor verdadero o falso, y determina si RIP intercambiará las rutas a través de la vif. La fijación de este comando en true permite que las rutas recibidas de una dirección IP sean temporalmente removidas sin borrar la CONFIGURACIÓN, el valor por defecto es falso.

Configuración de RIP en RTRLNX_1

```
/* RIP ipv4 RTRLNX_1 red 192.168 */
/*********************************/
/******Configuracion de RIP********/
/*********************************/
protocols { /*Inicio de configuracion del protocolo*/
      rip { /*Configuracion del protocolo RIP*/
      interface eth2 { /*Definicion de la interfaz por la que RIP
realizara el enrutamiento*/
      vif eth2 { /*Definicion de la interfaz virtual por la que RIP
realizara el enrutamiento*/
      address 20.1.1.2 { /*Definicion de la dirección IP de la interfaz
virtual por la que RIP realizara el enrutamiento*/
            disable:false /*Permite que no se elimine la configuración de
la dirección IP*/
                  } } }
      interface eth1 { /*Definicion de la interfaz por la que RIP
realizara el enrutamiento*/
      vif eth1 { /*Definicion de la interfaz virtual por la que RIP
realizara el enrutamiento*/
      address 10.1.1.2 { /*Definicion de la dirección IP de la interfaz
virtual por la que RIP realizara el enrutamiento*/
            disable:false /*Permite que no se elimine la configuración de
la dirección IP*/
                  } } } } }
```

3.3.2.7.2.2 OSPF

OSPF es uno de los principales protocolos de enrutamiento interior (IGP), OSPF versión 2 se utiliza para enrutamiento de paquetes IPv4, para paquetes IPv6 se utiliza OSPFv3.

En los archivos de configuración para configurar OSPFv2 la etiqueta utilizada es *ospf4* ya que se refiere a direcciones IPv4, la etiqueta para OSPFv3 es *ospf6* ya que se refiere a la direcciones.

Para ejecutar OSPF versión 2 o 3, el "router ID" debe ser especificado, se coloca una única dirección IPv4 en el sistema autónomo. OSPF divide redes en áreas, cada una debe ser configurada.

Se configura la interfaz/vif/dirección, en el caso OSPF versión 3 la dirección no es necesaria.

OSPF al ser un protocolo de enrutamiento interior se ejecuta en un único sistema autónomo. OSPF logra buenas propiedades de ampliación al dividir un AS en distintas regiones llamadas áreas. Las áreas se estructuran en una jerarquía de dos niveles, el área 0.0.0.0 llamada área de backbone y las demás áreas, que deben estar conectadas a la backbone, ya sea directamente o a través de enlaces virtuales.

Un aspecto fundamental en OSPF es que describe la topología de enrutamiento mediante los paquetes de Publicación de Estado de Enlace (LSA). Cada enrutador OSPF dentro de una zona debe tener exactamente el mismo LSA en su base de datos.

Las configuraciones del OSPFv2 y OSPFv3 son prácticamente equivalentes, con las siguientes excepciones:

OSPFv3 no admite la autenticación en el propio protocolo.

OSPFv2 soporta una sola dirección por la interfaz / vif, por lo tanto, todos los parámetros se fijan por debajo de la vif.

OSPFv3 soporta múltiples direcciones por interfaz / vif, por lo tanto se establecen los parámetros por debajo de la vif.

Los parámetros de configuración se utilizan de la siguiente manera:

protocols - protocolos: Delimita la configuración para todos los protocolos de enrutamiento en la configuración del enrutador XORP. Es obligatorio que la configuración de OSPF se encuentra bajo de este comando.

ospf4: delimita la parte de la configuración de OSPFv2 en el archivo de configuración del router XORP.

ospf6: delimita la parte de la configuración de OSPFv3 en el archivo de configuración del router XORP.

id del router - router-id: Representa el identificador del sistema autónomo. Se coloca, como una buena opción la dirección IP de una interfaz que pertenece al enrutador. El formato de la router-id es una dirección IPv4.

área: delimita el área en el que múltiples enlaces virtuales y las interfaces puedan ser configuradas. El área se especifica con la notación que se aplica para direcciones IPv4.

interface - interfaz: especifica una interfaz de red que debe utilizarse para el enrutamiento de OSPF. La interfaz debe estar configurada en la sección de interfaces del router.

vif: Especifica la interfaz virtual vif que se utiliza para el enrutamiento de OSPF.

dirección: especifica una dirección IPv4 que se debe utilizar para el enrutamiento de OSPF. OSPF se comunica con otros routers usando esta dirección.

Configuración de OSPF en RTRLNX_1

```
/* OSPF ipv4 RTRLNX_1 red 192.168 */
/*****************************************/
/*Configuracion para el reenvio de paquetes*/
/*****************************************/
fea {
    unicast-forwarding4 { /*habilita el reenvio para paquetes IPv4*/
        disable: false /*Permite que no se elimine la configuración*/
    } }
/*********************************/
/*****Configuracion de OSPF *******/
/*********************************/
protocols { /*Inicio de configuracion del protocolo*/
      ospf4 { /*Configuracion del protocolo OSPF*/
            router-id: 192.168.1.1 /*Identificador del AS*/
            area 0.0.0.0 { /*Configuracion del area de backbone */
```

```
                  interface eth1 { /*Definicion de la interfaz por la que
OSPF realizara el enrutamiento*/
                        vif eth1 { /*Definicion de la interfaz virtual
por la que OSPF realizara el enrutamiento*/
                              address 10.1.1.2 { /*Definicion de la
dirección IP de la interfaz virtual por la que OSPF realizara el
enrutamiento*/                                  } } }
                  interface eth2 { /*Definicion de la interfaz por la que
OSPF realizara el enrutamiento*/
                        vif eth2 { /*Definicion de la interfaz virtual
por la que OSPF realizara el enrutamiento*/
                              address 20.1.1.2 { /*Definicion de la
dirección IP de la interfaz virtual por la que OSPF realizara el
enrutamiento*/                                  } } }
                                    } } } }
            area 0.0.0.1 { /*Configuracion del area local */
                  interface eth0 { /*Definicion de la interfaz local*/
                        vif eth0 { /*Definicion de la interfaz virtual
local*/
                              address 192.168.1.1 { /*Definicion de la
dirección IP de la interfaz virtual local*/
                                    } } } } } }
```

3.3.2.7.2.3 BGP

BGP es el principal protocolo de enrutamiento de dominio en Internet. BGP versión 4 se especifica en el RFC 4271, XORP BGP es compatible con el nuevo RFC.

El principal concepto utilizado en BGP es el de Sistema Autónomo. Un AS corresponde a un dominio de enrutamiento que se encuentra bajo una autoridad administrativa, y que implementa su propia política de enrutamiento.

Para ejecutar BGP el "bgp-id" (BGP Identifier) y "local-as" (Número de sistema autónomo) deben ser especificados. Los parámetros de configuración se utilizan de la siguiente manera:

protocols - protocolos: delimita la configuración para todos los protocolos de enrutamiento en el archivo de configuración del router XORP. Es obligatorio que BGP se encuentre bajo este comando.

bgp: delimita la parte de configuración del protocolo BGP en el archivo de configuración del router XORP.

bgp-id: es el identificador del protocolo BGP. Normalmente se establece una de las direcciones IP del router. El formato de este ID es similar al formato empleado

en direcciones IPv4. Este identificador es necesario incluso si el router sólo es compatible con el reenvío de paquetes IPv6.

local as - sistema autónomo local: este es el número de sistema autónomo.

peer - pares: este comando delimita la configuración de una asociación BGP con otro router. Este identificador es la dirección de la red vecina.

- *ip local – Dirección IP local:* es la dirección IP de la interfaz ethernet del enrutador, esta dirección se utiliza para el establecimiento de las conexiones BGP.
- *as:* es el número de AS, debe coincidir con el número que se coloca en *peer*, es obligatorio especificarlo.
- *next hop – próximo salto:* es la dirección IPv4, que representa el siguiente salto a dar por los paquetes a ser transmitidos.
- *ipv4-unicast:* Toma un valor verdadero o falso, y especifica si debe negociar BGP para permitir el intercambio de direcciones IPv4. Es activado por defecto.

Configuración de BGP en RTRLNX_1

```
/* BGP ipv4 RTRLNX_1 red 192.168 */
/*************************************************/
/*****Configuracion para el reenvio de paquetes *******/
/*************************************************/
fea {
    unicast-forwarding4 { /*habilita el reenvio para paquetes IPv4*/
        disable: false /*Permite que no se elimine la configuración*/
    } }
/*********************************/
/*****Configuracion de BGP *******/
/*********************************/
protocols { /*Inicio de configuracion del protocolo*/
    bgp {  /* configuracion del protocolo BGP*/
        bgp-id: 192.168.1.0 /*Identificador del protocolo*/
        local-as: 1000 /* Numero del sistema autonomo local*/
            peer 172.16.1.0 { /* Direccion de la red vecina*/
            local-ip: 10.1.1.2 /*Direccion IP de la interfaz ethernet
eth1*/
            as: 1000 /* Numero del sistema autonomo local*/
            next-hop: 10.1.1.2  /*Direccion IP de la interfaz ethernet
eth1*/
            ipv4-unicast: true /* Permite el intercambio de direcciones
IPv4 a traves de BGP*/
          }
```

```
        peer 100.100.1.0 { /* Direccion de la red vecina*/
            local-ip: 20.1.1.2 /*Direccion IP de la interfaz ethernet
eth2*/
            as: 1000 /* Numero del sistema autonomo local*/
            next-hop: 20.1.1.2 /*Direccion IP de la interfaz ethernet
eth2*/
            ipv4-unicast: trae /* Permite el intercambio de direcciones
IPv4 a traves de BGP*/
          } } }
```

3.3.2.8 DIRECCIONAMIENTO IP VERSIÓN 6

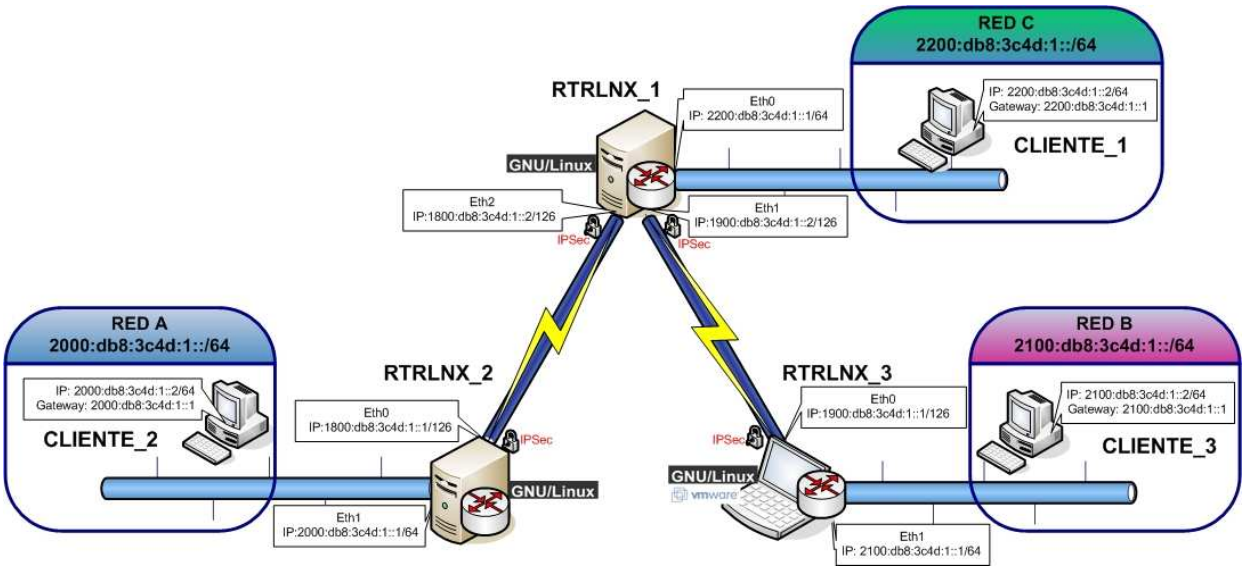

Figura 3- 3 Direccionamiento IPv6

Configuración de interfaces

Los comandos para configurar las interfaces son los mismos que IPv4, cada uno fue explicado en la sección anterior, la variación más significativa es en las direcciones IP.

Se establece una dirección adicional para cada interfaz, esta dirección permite comunicación con otros enrutadores. Esta se denomina dirección de enlace local.

Configuración de interfaces en RTRLNX_1

```
/*Enrutamiento Estatico IPv6 RTRLNX Red 192.168 */
rtrmgr {
```

```
    config-directory: "/etc/xorp" /* Ubicación del archivo del
configuracion*/
}
interfaces { /*Inicio de configuracion de las interfaces */
/*******************************/
/*Configuracion de la Ethernet 0*/
/*******************************/
 interface eth0 { /*Configuracion de la Interfaz Ethernet0*/
        vif eth0 { /*Configuración de la interfaz virtual para la
ethernet0*/
            address 2200:0db8:3c4d:0001:0000:0000:0000:0001  {
/*Configuracion de la direccion IPv6*/
                prefix-length: 64 /*Asignacion de la longitud del
prefijo*/
            }
            address fe80::20c:29ff:fecb:62fe { /* Direccion de enlace
local para la comunicacion con el router vecino*/
                prefix-length: 64 /*Asignacion de la longitud del
prefijo*/
            } } }
/*******************************/
/*Configuracion de la Ethernet 1*/
/*******************************/
    interface eth1 { /*Configuracion de la Interfaz Ethernet1*/
        vif eth1 { /*Configuración de la interfaz virtual para la
ethernet1*/
            address 1900:0db8:3c4d:0001:0000:0000:0000:0002 {
/*Configuracion de la direccion IPv6*/
                prefix-length: 64 /*Asignacion de la longitud del
prefijo*/
            }
            address fe80::20c:29ff:fecb:6208 { /* Direccion de enlace
local para la comunicacion con el router vecino*/
                prefix-length: 64 /*Asignacion de la longitud del
prefijo*/
            } } }
/*******************************/
/*Configuracion de la Ethernet 2*/
/*******************************/
 interface eth2 { /*Configuracion de la Interfaz Ethernet2*/
        vif eth2 { /*Configuración de la interfaz virtual para la
ethernet2*/
            address 1800:0db8:3c4d:0001:0000:0000:0000:0002 {
/*Configuracion de la direccion IPv6*/
                prefix-length: 64 /*Asignacion de la longitud del
prefijo*/
            }
            address fe80::20c:29ff:fecb:6212 { /* Direccion de enlace
local para la comunicacion con el router vecino*/
                prefix-length: 64 /*Asignacion de la longitud del
prefijo*/
            } } } }
```

3.3.2.8.1 Rutas estáticas

Configuración de rutas estáticas en RTRLNX_1

```
/*Enrutamiento Estatico IPv6 RTRLNX Red 192.168 */
/*****************************************/
/*Configuracion para el reenvio de paquetes*/
/*****************************************/
fea {
    unicast-forwarding6 { /*Habilita el reenvio para paquetes IPv6*/
      disable: false /*Permite que no se elimine la configuración*/
    } }
/*********************************************/
/******Configuracion de rutas estaticas********/
/*********************************************/
protocols {
static { /*Configuracion de rutas estaticas*/
route 2000:0db8:3c4d:0001:0000:0000:0000:0000/64 { /*Dirección de la RED
A, a ser alcanzada*/
      next-hop: fe80::20c:29ff:fe4c:ab8e /*Dirección del link local de la
eth0 del router A*/
        metric: 1 /*Definición de un salto realizado por el paquete para
alcanzar la RED A*/
           }
route 2100:0db8:3c4d:0001:0000:0000:0000:0000/64 { /*Dirección de la RED
B, a ser alcanzada*/
      next-hop: fe80::20c:29ff:feb9:3f88 /*Dirección del link local de la
eth0 del router B*/
        metric: 1 /*Definición de un salto realizado por el paquete para
alcanzar la RED B*/
           } } }
```

3.3.2.8.2 Protocolo De Enrutamiento Dinámico

3.3.2.8.2.1 RIPng

La configuración para RIPng es básicamente la misma que RIP, con la excepción que las direcciones son IPv6.

Configuración de RIPng en RTRLNX_1

```
/*RIP IPv6 RTRLNX Red 2200 */
/*****************************************/
/*Configuracion para el reenvio de paquetes*/
/*****************************************/
fea {
    unicast-forwarding6 { /*Habilita el reenvio para paquetes IPv6*/
      disable: false /*Permite que no se elimine la configuración*/
    } }
/*********************************/
/******Configuracion de RIPng********/
/*********************************/
protocols { /*Inicio de configuracion del protocolo*/
    ripng { /*Configuracion del protocolo RIPng*/
```

```
      interface eth1 { /*Definicion de la interfaz por la que RIPng
realizara el enrutamiento hacia la RED B*/
          vif eth1 { /*Definicion de la interfaz virtual por la que RIPng
realizara el enrutamiento*/
            address fe80::20c:29ff:fecb:6208 { /*Definicion del link
local de la  eth1 del router C por la que realizara el enrutamiento hacia
la RED B */
                disable: false /*Permite que no se elimine la
configuración de la dirección IPv6*/
            } } }
      interface eth2 { /*Definicion de la interfaz por la que RIPng
realizara el enrutamiento hacia la RED A*/
          vif eth2 { /*Definicion de la interfaz virtual por la que RIPng
realizara el enrutamiento*/
            address fe80::20c:29ff:fecb:6212 { /*Definicion del link
local de la  eth2 del router C por la que realizara el enrutamiento hacia
la RED A */
                disable: false /*Permite que no se elimine la
configuración de la dirección IPv6*/
            } } } } }
```

3.3.2.8.2.2 OSPF

Configuración de OSPF en RTRLNX_1

```
/*OSPF IPv6 RTRLNX Red 2000 */
/*****************************************/
/*Configuracion para el reenvio de paquetes*/
/*****************************************/
fea {
    unicast-forwarding6 { /*Habilita el reenvio para paquetes IPv6*/
      disable: false /*Permite que no se elimine la configuración*/
          } }
/*********************************/
/******Configuracion de OSPFv6*******/
/*********************************/
protocols { /*Inicio de configuracion del protocolo*/
      ospf6  { /*Configuracion del protocolo OSPF*/
            router-id: 192.168.1.1 /*Identificador del AS*/
            area 0.0.0.0 { /*Configuracion del area de backbone */
                 interface eth1 { /*Definicion de la interfaz por la que
OSPF realizara el enrutamiento*/
                     vif eth1 { /*Definicion de la interfaz virtual por
la que OSPF realizara el enrutamiento*/
                     address fe80::20c:29ff:fecb:6208 { /*Definicion del
link local de la  eth1 del router C por la que realizara el enrutamiento
hacia la RED B */
                     disable: false /*Permite que no se elimine la
configuración de la dirección IPv6*/
            } } }
                  interface eth2 {
                      vif eth2 { /*Definicion de la interfaz virtual por
la que OSPF realizara el enrutamiento*/
```

```
                    address fe80::20c:29ff:fecb:6212 { /*Definicion del
link local de la  eth2 del router C por la que realizara el enrutamiento
hacia la RED A */
                    disable: false /*Permite que no se elimine la
configuración de la dirección IPv6*/
          } } } } } }
```

3.3.2.8.2.3 BGP

Configuración de BGP en RTRLNX_1

```
/* BGP ipv6 RTRLNX_1 red 2000 */
/*****************************************/
/*Configuracion para el reenvio de paquetes*/
/*****************************************/
fea {
    unicast-forwarding6 { /*Habilita el reenvio para paquetes IPv6*/
      disable: false /*Permite que no se elimine la configuración*/
         } }
/***********************************/
/******Configuracion de BGP*******/
/***********************************/
protocols { /*Inicio de configuracion del protocolo*/
    bgp { /* configuracion del protocolo BGP*/
        bgp-id: 192.168.1.0 /*Identificador del protocolo*/
        local-as: 1000 /* Numero del sistema autonomo local*/
            peer 2100:0db8:3c4d:0001:0000:0000:0000:0001{ /* Direccion de
la red B*/
            local-ip: 1900:0db8:3c4d:0001:0000:0000:0000:0002 /*Direccion
IP de la interfaz ethernet eth1*/
            as: 1000 /* Numero del sistema autonomo local*/
            next-hop: 1900:0db8:3c4d:0001:0000:0000:0000:0002 /*Direccion
IP de la interfaz ethernet eth1*/
            ipv6-unicast: true /* Permite el intercambio de direcciones
IPv6 a traves de BGP*/
         }
        peer 2000:0db8:3c4d:0001:0000:0000:0000:0001{ /* Direccion de la
red C*/
            local-ip: 1800:0db8:3c4d:0001:0000:0000:0000:0002 /*Direccion
IP de la interfaz ethernet eth2*/
            as: 1000 /* Numero del sistema autonomo local*/
            next-hop: 1800:0db8:3c4d:0001:0000:0000:0000:0002 /*Direccion
IP de la interfaz ethernet eth2*/
            ipv6-unicast: true /* Permite el intercambio de direcciones
IPv6 a traves de BGP*/
         } } }
```

3.3.3 SOFTWARE DE CALIDAD DE SERVICIO

De acuerdo a la selección de Script para implementación de Calidad de Servicio se ha obtenido como resultado CBQ.Init

CBQ (Class Based Queueing, Encolamiento Basado en Clases), cbq es básicamente un Script escrito en BASH que permite realizar la gestión y control del ancho de banda en GNU/Linux. Fue creado originalmente en 1999 por Pavel Golubev y posteriormente mantenido desde el 2001 hasta el 2004 por Lubomir Bulej. Este script utiliza básicamente de una forma simplificada los mandatos IP y TC para su funcionamiento, estos comandos forman parte del paquete iproute, que incluye en los instaladores de la mayor parte de las distribuciones de GNU/Linux.

El script CBQ.Init, tiene el nombre de shaper para la distribución Debían.

3.3.3.1 **Objetivo**

El objetivo principal es controlar el ancho de banda y la asignación de prioridades a los paquetes transmitidos en puertos específicos, redes o direcciones IP y limitarlos de acuerdo a las necesidades impuestas.

3.3.3.2 **Requisito**

El requisito indispensable es tener instalado el paquete iproute2.

3.3.3.3 **Instalación**

Este paquete se lo puede encontrar fácilmente en los repositorios oficiales de Debian, por medio del comando:

```
apt-get install shaper
```

En el caso de poseer el paquete .deb se procede con los comandos:

```
dpkg –i shaper2.2.12_all
```

3.3.3.4 **Configuración**

La configuración se debe realizar creando archivos por cada clase de tráfico, los archivos se deben describir en el CBQ_PATH en el directorio ubicado /etc/shaper/.

Los nombres de los archivos de configuración deben poseer un nombre que deben cumplir con un formato obligatorio: **cbq-<clsid>.<name>**

<clsid>: Es un número hexadecimal de dos bytes dentro del rango <0002-FFFF> que identifica a cada clase.

<name>: Es el nombre de la clase que permite distinguir el archivo de configuración.

Para cada configuración de clase, se deben asignar valores al menos para los parámetros obligatorios.

Para construir las reglas, se requiere al menos comprender y especificar los valores para los parámetros **DEVICE, WEIGHT, RATE y RULE**.

- Parámetro DEVICE.

Es un parámetro obligatorio. Se determina los valores con el nombre de la interfaz, ancho de banda y peso de esta interfaz. Este último valor, que es opcional en este parámetro, se calcula dividiendo el ancho de banda de la interfaz para diez. Por ejemplo, sí se dispone de una interfaz denominada eth0 de 100 Mbps, el peso será 10 Mbit/s, de tal modo los valores del parámetro DEVICE, quedarían de la siguiente forma:

```
DEVICE=eth0,100Mbit,10Mbit
```

- Parámetro RATE.

Es un parámetro obligatorio, que define el ancho de banda que será asignado a la clase. El tráfico que pase a través de esta clase será modificado para ajustarse a

la proporción definida. Por ejemplo, sí se quiere limitar el ancho de banda utilizado a 10 Mbps, el valor de RATE sería 10Mbit, como se muestra a continuación.

```
RATE=10Mbit
```

- Parámetro WEIGHT.

Es un parámetro obligatorio. Éste es proporcional al ancho de banda total de la interfaz. Como regla se calcula dividiendo entre diez el ancho de banda total de la interface. Para un ancho de banda de 2048 Kbps, correspondería un valor de 204 Kbit:

```
WEIGHT=204Kbit
```

- Parámetro PRIO.

Es un parámetro opcional, que se utiliza para especificar que prioridad tendrá sobre otras reglas de control de ancho de banda. Mientras más alto sea el valor, menos prioridad tendrá sobre otras reglas. Se recomienda utilizar el valor 5 que funcionará para la mayoría de los casos. Ejemplo:

```
PRIO=5
```

- Parámetros RULE.

Es un parámetro obligatorio. Son las reglas de filtración que se utilizan para seleccionar tráfico en cada una de las clases. La sintaxis completa es la siguiente:

```
RULE=[[saddr[/prefijo]][:puerto[/máscara]],][daddr[/prefijo]][:puerto[/máscara]]
```

saddr: se refiere a la dirección de origen.

daddr: se refiere a la dirección de destino.

La sintaxis es la siguiente, donde todos los valores son opcionales, pero se debe especificar al menos uno:

```
RULE=IP-origen:puerto-origen,IP-destino:puerto-destino
```

La interpretación sigue cuatro simples principios:

- Cualquier dirección IP o red que se coloque antes de la coma se considera dirección IP o red de origen.
- Cualquier dirección IP o red que se coloque después de la coma se considera dirección IP o red de destino.
- Cualquier puerto antes de la coma se considera el puerto de origen.
- Cualquier puerto especificado después de la coma se considera puerto de destino.

Ejemplos.

Selección de todo el tráfico desde cualquier puerto en cualquier red hacia los puertos 25 (SMTP), 465 (SMTPS) y 587 (SMTP) en cualquier red (es decir, controla ancho de banda de correo saliente):

```
RULE=,:25
RULE=,:465
RULE=,:587
```

Selección de todo el tráfico desde los puertos 25 (SMTP), 465 (SMTPS) y 587 (SMTP) en cualquier red hacia cualquier puerto en cualquier red (es decir, controla ancho de banda de correo entrante):

```
RULE=:25,
RULE=:465,
RULE=:587,
```

Selección de todo el tráfico desde la red 192.168.0.0/24 hacia cualquier puerto en cualquier red:

```
RULE=192.168.0.0/24,
```

Selección de todo el tráfico desde cualquier puerto en cualquier red hacia cualquier puerto en la red 192.168.0.0/24:

```
RULE=,192.168.0.0/24
```

Selección de todo el tráfico desde cualquier puerto en la red 192.168.0.0/24 hacia el puerto 25 (SMTP) en cualquier red:

```
RULE=192.168.0.0/24,:25
```

Selección de todo el tráfico desde el puerto 25 (SMTP) en la red 192.168.0.0/24 hacia cualquier puerto en cualquier red:

```
RULE=192.168.0.0/24:25,
```

Un archivo de configuración válido quedaría de la siguiente forma:

Contenido de fichero /etc/shaper/cbq-0002.smtp-in:

```
DEVICE=eth0,2048Kbit
RATE=512Kbit
WEIGHT=204Kbit
PRIO=5
RULE=:25,192.168.0.0/24
RULE=:465,192.168.0.0/24
RULE=:587,192.168.0.0/24
```

3.3.3.5 **Administración del Servicio**

Para probar que las clases están correctas antes de utilizar éstas, puede utilizar el comando:

```
/etc/init.d/shaper compile
```

Para ejecutar por primera e iniciar el servicio cbq, se debe utilizar, por medio del comando:

```
/etc/init.d/shaper start
```

Para hacer que los cambios hechos tras modificar la configuración tengan efecto, se debe utilizar:

```
/etc/init.d/shaper restart
```

Para detener el servicio cbq y eliminar de memoria todas las reglas se debe utilizar:

```
/etc/init.d/shaper stop
```

Para supervisar las estadísticas de tráfico gestionado a través de cbq se debe utilizar:

```
/etc/init.d/shaper stats
```

3.3.3.6 **Archivos de Configuración Calidad de Servicio**

Configuración RTRLNX 1

- **cbq-0100-172eth0in,** regla para control de ancho de banda de entrada de tráfico por la interfaz eth0 desde la red B hacia el resto de redes.

```
DEVICE=eth0,1000Mbit,100Mbit  /*Interfaz de red eth0,
                        Capacidad de 1Gbps, Peso de 100Mbps*/
RATE=2Mbit              /*Ancho de banda asignado 2Mbps */
WEIGHT=200bit           /* Peso 2Mbps */
PRIO=5                  /* Prioridad intermedia 5/10 */
RULE=172.16.1.0/24,     /* Regla aplica de la red 172.16.1.0
                            hacia cualquier otra red*/
LEAF=sfq                /* Disciplina de Encolamiento de
                            Trafico*/
BOUNED=yes              /* La clase no puede pedir prestado
                            ancho de banda a la clase padre*/
```

- **cbq-0200-172eth0out,** regla para control de ancho de banda de salida de tráfico por la interfaz eth0 desde la red B hacia el resto de redes.

```
DEVICE=eth0,1000Mbit,100Mbit /*Interfaz de red eth0,
                        Capacidad de 1Gbps, Peso de 100Mbps*/
RATE=2Mbit              /*Ancho de banda asignado 2Mbps */
WEIGHT=200bit           /* Peso 2Mbps */
PRIO=5                  /* Prioridad intermedia 5/10 */
RULE=,172.16.1.0/24     /* Regla aplica desde cualquier red
                            hacia la red 172.16.1.0 */
LEAF=sfq                /* Disciplina de Encolamiento de
                            Trafico*/
BOUNED=yes              /* La clase no puede pedir prestado
                            ancho de banda a la clase padre*/
```

- **cbq-0300-172eth1in,** regla para control de ancho de banda de entrada de tráfico por la interfaz eth1 desde la red B hacia el resto de redes

```
DEVICE=eth1,1000Mbit,100Mbit
RATE=2Mbit
WEIGHT=200bit
PRIO=5
RULE=172.16.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-0400-172eth1out,** regla para control de ancho de banda de salida de tráfico por la interfaz eth1 desde la red B hacia el resto de redes.

```
DEVICE=eth1,1000Mbit,100Mbit
RATE=2Mbit
WEIGHT=200bit
```

```
PRIO=5
RULE=,172.16.1.0/24
LEAF=sfq
BOUNED=yes
```

- **cbq-0500-172eth2in**, regla para control de ancho de banda de entrada de tráfico por la interfaz eth2 desde la red B hacia el resto de redes.

```
DEVICE=eth2,1000Mbit,100Mbit
RATE=2Mbit
WEIGHT=200bit
PRIO=5
RULE=172.16.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-0600-172eth2out,** regla para control de ancho de banda de salida de tráfico por la interfaz eth2 desde la red B hacia el resto de redes.

```
DEVICE=eth2,1000Mbit,100Mbit
RATE=2Mbit
WEIGHT=200bit
PRIO=5
RULE=,172.16.1.0/24
LEAF=sfq
BOUNED=yes
```

- **cbq-0700-100eth0in,** regla para control de ancho de banda de entrada de tráfico por la interfaz eth0 desde la red A hacia el resto de redes.

```
DEVICE=eth0,1000Mbit,100Mbit
RATE=6Mbit
WEIGHT=600bit
PRIO=5
RULE=192.168.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-0800-100eth0out,** regla para control de ancho de banda de salida de tráfico por la interfaz eth0 desde la red A hacia el resto de redes.

```
DEVICE=eth0,1000Mbit,100Mbit
RATE=6Mbit
WEIGHT=600bit
PRIO=5
```

```
RULE=,192.168.1.0/24
LEAF=sfq
BOUNED=yes
```

- **cbq-0900-100eth1in,** regla para control de ancho de banda de entrada de tráfico por la interfaz eth1 desde la red A hacia el resto de redes.

```
DEVICE=eth1,1000Mbit,100Mbit
RATE=6Mbit
WEIGHT=600bit
PRIO=5
RULE=192.168.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-1000-100eth1out**, Regla para control de ancho de banda de salida de tráfico por la interfaz eth1 desde la red A hacia el resto de redes.

```
DEVICE=eth1,1000Mbit,100Mbit
RATE=6Mbit
WEIGHT=600bit
PRIO=5
RULE=,192.168.1.0/24
LEAF=sfq
BOUNED=yes
```

- **cbq-1100-100eth2in,** regla para control de ancho de banda de entrada de tráfico por la interfaz eth2 desde la red A hacia el resto de redes.

```
DEVICE=eth2,1000Mbit,100Mbit
RATE=6Mbit
WEIGHT=600bit
PRIO=5
RULE=192.168.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-1200-100eth2out,** regla para control de ancho de banda de salida de tráfico por la interfaz eth2 desde la red A hacia el resto de redes.

```
DEVICE=eth2,1000Mbit,100Mbit
RATE=6Mbit
WEIGHT=600bit
PRIO=5
RULE=,192.168.1.0/24
LEAF=sfq
```

```
BOUNED=yes
```

- **cbq-1300-192eth0in,** regla para control de ancho de banda de entrada de tráfico por la interfaz eth0 desde la red C hacia el resto de redes.

```
DEVICE=eth0,1000Mbit,100Mbit
RATE=4Mbit
WEIGHT=400bit
PRIO=5
RULE=100.100.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-1400-192eth0out**, regla para control de ancho de banda de salida de tráfico por la interfaz eth0 desde la red C hacia el resto de redes.

```
DEVICE=eth0,1000Mbit,100Mbit
RATE=4Mbit
WEIGHT=400bit
PRIO=5
RULE=,100.100.1.0/24
LEAF=sfq
BOUNED=yes
```

- **cbq-1500-192eth1in,** regla para control de ancho de banda de entrada de tráfico por la interfaz eth1 desde la red C hacia el resto de redes.

```
DEVICE=eth1,1000Mbit,100Mbit
RATE=4Mbit
WEIGHT=400bit
PRIO=5
RULE=100.100.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-1600-192eth1out,** regla para control de ancho de banda de salida de tráfico por la interfaz eth1 desde la red C hacia el resto de redes.

```
DEVICE=eth1,1000Mbit,100Mbit
RATE=4Mbit
WEIGHT=400bit
PRIO=5
RULE=,100.100.1.0/24
LEAF=sfq
BOUNED=yes
```

- **cbq-1700-192eth2in**, regla para control de ancho de banda de entrada de tráfico por la interfaz eth2 desde la red C hacia el resto de redes.

```
DEVICE=eth2,1000Mbit,100Mbit
RATE=4Mbit
WEIGHT=400bit
PRIO=5
RULE=100.100.1.0/24,
LEAF=sfq
BOUNED=yes
```

- **cbq-1800-192eth2out,** regla para control de ancho de banda de salida de tráfico por la interfaz eth2 desde la red C hacia el resto de redes.

```
DEVICE=eth2,1000Mbit,100Mbit
RATE=4Mbit
WEIGHT=400bit
PRIO=5
RULE=,100.100.1.0/24
LEAF=sfq
BOUNED=yes
```

3.3.4 INSTALACIÓN Y CONFIGURACION DE IPSEC

IPsec es una extensión del protocolo IP, proporciona servicios criptográficos de seguridad, permitiendo autenticación, integridad, y confidencialidad.

IPsec proporciona encriptación y autenticación a nivel de red, dando lugar a una solución de seguridad.

3.3.4.1 **Modos de Funcionamiento**

3.3.4.1.1 Modo Túnel

Este modo se usa cuando uno de los extremos de la comunicación es un gateway, se encripta el paquete IP en su totalidad, convirtiendo el resultado en los datos de un nuevo paquete IP. El enrutador de origen encripta el paquete y lo

reenvía por el túnel IPsec, el enrutador de destino desencripta el paquete IP original y lo reenvía al sistema destino.

3.3.4.1.2 Modo Transporte

Se usa cuando es un host el que genera los paquetes, solo se encriptan los datos, y la cabecera IP original se deja intacta este modo tiene la ventaja de que añade pocos bytes a cada paquete además permite a los dispositivos de la red pública ver el origen y el destino del paquete, esto también implica que un atacante que esté realizando un análisis del tráfico que circula por la red pueda ver el origen y destino del mismo.

3.3.4.2 **Protocolos de IPsec**

IPsec emplea dos protocolos diferentes - AH y ESP - para asegurar la autenticación, integridad y confidencialidad de la comunicación. Puede proteger el datagrama IP completo o sólo los protocolos de capas superiores.

Authentication Header (AH) cabecera de autenticación, proporciona integridad, autenticación y no repudio si se eligen los algoritmos criptográficos apropiados.

Encapsulating Security Payload (ESP) protocolo de encapsulado de datos seguros, proporciona confidencialidad es una opción altamente recomendable de autenticación y protección de integridad.

Los algoritmos criptográficos definidos para usar con IPsec incluyen HMAC- SHA1 para protección de integridad, y Triple DES-CBC y AES para confidencialidad.

3.3.4.3 **Protocolo IKE**

El protocolo IKE resuelve el problema más importante del establecimiento de comunicaciones seguras: la autenticación de los participantes y el intercambio de claves simétricas. Tras ello, crea las asociaciones de seguridad. El protocolo IKE suele implementarse a través de servidores de espacio de usuario, y no suele

implementarse en el sistema operativo. El protocolo IKE emplea el puerto 500 UDP para su comunicación.

El protocolo IKE funciona en dos fases: la primera fase establece un isakmp sa (asociación de seguridad del protocolo de gestión de claves de Internet).

En la segunda fase, el ISAKMP SA se emplea para negociar y establecer las asociaciones de seguridad de IPsec.

3.3.4.4 **Configuración de IPsec en Linux**

Lo primero es instalar el paquete de IPsec, mediante el comando:

#apt-get install ipsec-tools

El archivo de configuración que se crea al instalar el paquete se llama ipsec-tools.conf y está ubicado en el directorio /etc. Por defecto el archivo contiene las siguientes líneas:

```
#!/usr/sbin/setkey -f
# NOTE: Do not use this file if you use racoon with racoon-tool
# utility. racoon-tool will setup SAs and SPDs automatically using
# /etc/racoon/racoon-tool.conf configuration.
## Flush the SAD and SPD
#flush;
#spdflush;
## Some sample SPDs for use racoon
# spdadd 10.10.100.1 10.10.100.2 any -P out ipsec
# esp/transport//require;
# spdadd 10.10.100.2 10.10.100.1 any -P in ipsec
# esp/transport//require;
#
```

Los parámetros que deben ser modificados en el archivo de configuración son:

Primero se debe eliminar los numerales para habilitar los parámetros

flush;

spdflush;

Se debe recordar que IPsec soporta dos protocolos que son AH (AUTHENTICATION HEADER) y ESP (ENCAPSULATION SECURITY PAYLOAD) y a continuación se muestra la configuración de IPsec con cada uno y con ambos.

3.3.4.4.1 Configuración de IPsec con AH

AH es un protocolo que proporciona en el ámbito de IPsec la autenticación del emisor y la integridad del mensaje mediante el cálculo de un código HMAC.

Para que IPsec trabaje con AH se debe agregar las siguientes líneas en el archivo de configuración (las direcciones IP utilizadas aquí son para el caso del ejemplo, para otra configuración se debe tener en cuenta el rango de direccionamiento de la red):

add 192.168.0.10 192.168.0.20 ah 0x200 -A hmac-md5 "abcdefghijklmnop";

add 192.168.0.20 192.168.0.10 ah 0x300 -A hmac-md5 "abcdefghijklmnop";

192.168.0.10 es la dirección IP local y 192.168.0.20 es la dirección del equipo con el que se establecerá la conexión IPsec, hmac-md5 es el algoritmo que se usará y "abcdefghijklmnop" es la clave pre compartida que debe ser de 1024 bits ya que esto es lo que soporta md5. La longitud varía dependiendo del algoritmo que se utilice.

La segunda línea se agrega para establecer la conexión en sentido contrario, estableciendo así la comunicación en doble sentido.

También se debe modificar las siguientes líneas:

spdadd 192.168.0.10 192.168.0.20 any -P out ipsec

ah/transport//require;

spdadd 192.168.0.20 192.168.0.10 any -P in ipsec

ah/transport//require;

Con esto se especifica nuevamente la comunicación en ambos sentidos y también que se trabaje con AH.

Así queda el archivo de configuración:

```
#!/usr/sbin/setkey -f
# /etc/racoon/racoon-tool.conf configuration.
## Flush the SAD and SPD
flush;
spdflush;
## Some sample SPDs for use racoon
add 192.168.0.10 192.168.0.20 ah 0x200 -A hmac-md5 "abcdefghijklmnop";
add 192.168.0.20 192.168.0.10 ah 0x300 -A hmac-md5 "abcdefghijklmnop";
spdadd 192.168.0.10 192.168.0.20 any -P out ipsec
ah/transport//require;
spdadd 192.168.0.20 192.168.0.10 any -P in ipsec
ah/transport//require;
#
```

Algo que se debe tener en cuenta es que en el equipo con el que se va a establecer la conexión IPsec debe tener los mismos parámetros configurados pero en sentido contrario al local, por ejemplo, veamos cómo se vería el archivo de configuración del equipo 192.168.0.20:

```
#!/usr/sbin/setkey -f
# /etc/racoon/racoon-tool.conf configuration.
#
## Flush the SAD and SPD
#
flush;
spdflush;
```

```
## Some sample SPDs for use racoon
add 192.168.0.20 192.168.0.10 ah 0x300 -A hmac-md5 "abcdefghijklmnop";
add 192.168.0.10 192.168.0.20 ah 0x200 -A hmac-md5 "abcdefghijklmnop";
#
spdadd 192.168.0.20 192.168.0.10 any -P out ipsec
ah/transport//require;
#
spdadd 192.168.0.10 192.168.0.20 any -P in ipsec
ah/transport//require;
```

Después de tener listos los archivos de configuración recargamos IPsec

#setkey -f /etc/ipsec-tools.conf

Ahora se puede comprobar si está funcionando la encripción capturando paquetes con el comando tcpdump.

Si la configuración esta correcta se deben observar la transmisión de paquetes encriptados.

3.3.4.4.2 Configuración de IPsec con ESP

ESP es un protocolo que proporciona en el ámbito IPsec confidencialidad, autenticación y protección de integridad utilizando llaves cifradas.

El archivo de configuración queda de la siguiente forma:

```
#!/usr/sbin/setkey -f
# /etc/racoon/racoon-tool.conf configuration.
## Flush the SAD and SPD
flush;
spdflush;
## Some sample SPDs for use racoon
```

```
add 192.168.0.10 192.168.0.20 esp 0x201 -E 3des-cbc
"abcdefghijklmnopqrstuvwx";

add 192.168.0.20 192.168.0.10 esp 0x301 -E 3des-cbc
"abcdefghijklmnopqrstuvwx";

#

spdadd 192.168.0.10 192.168.0.20 any -P out ipsec

esp/transport//require;

#

spdadd 192.168.0.20 192.168.0.10 any -P in ipsec

esp/transport//require;

#
```

Las diferencias principales en comparación con la configuración de AH son: El algoritmo de cifrado cambia de md5 a 3des y por eso la longitud de la clave pre compartida cambia a 2048 bits, es decir 24 caracteres y que ahora la opción "-A" cambio a "-E" para encriptar.

Se recarga IPsec.

#setkey -f /etc/ipsec-tools.conf

Luego se realiza la misma prueba que se uso con AH, es decir usando tcpdump.

Si la configuración es correcta nos debe mostrar los paquetes encriptados.

3.3.4.4.3 Configuración de IPsec con AH y ESP

Es posible que IPsec funcione con AH y ESP juntos para hacerlo mucho más seguro. Para hacerlo simplemente se juntan las dos configuraciones realizadas anteriormente, como se muestra en el archivo de configuración siguiente:

```
#!/usr/sbin/setkey -f

# /etc/racoon/racoon-tool.conf configuration.

## Flush the SAD and SPD

flush;
```

```
spdflush;

## Some sample SPDs for use racoon

add 192.168.0.10 192.168.0.20 ah 0x200 -A hmac-md5 "abcdefghijklmnop";

add 192.168.0.20 192.168.0.10 ah 0x300 -A hmac-md5 "abcdefghijklmnop";

add 192.168.0.10 192.168.0.20 esp 0x201 -E 3des-cbc
"abcdefghijklmnopqrstuvwx";

add 192.168.0.20 192.168.0.10 esp 0x301 -E 3des-cbc
"abcdefghijklmnopqrstuvwx";

#

spdadd 192.168.0.10 192.168.0.20 any -P out ipsec

esp/transport//require

ah/transport//require;

#

spdadd 192.168.0.20 192.168.0.10 any -P in ipsec

esp/transport//require

ah/transport//require;

#
```

Se recarga IPsec.

#setkey -f /etc/ipsec-tools.conf

Y al capturar el tráfico con tcpdump muestra un resultado con paquetes encriptados.

3.3.4.5 **Archivos de Configuración IPv4**

Configuración RTRLNX 1

```
# IPSec IPv4 RTRLNX 192.168
global:
log: notify
connection(192-172):                /*Generacion de conexion*/
      src_range: 192.168.1.0/24     /*Red local en la que se va a
                                      encriptar la información */
```

```
      dst_range: 172.16.1.0/24      /*Red remota con la que se va a
                                       encriptar la información */
      src_ip: 192.168.1.1           /*Direccion del enrutador local*/
      dst_ip: 172.16.1.1            /*Direccion del enrutador remoto*/
      admin_status: yes             /*Conexión Administrada*/
      authentication_algorithm: hmac_sha1  /* Se utilizara el protocolo
                                          de autenticación hmac_sha1 */
      encryption_algorithm: 3des    /* Se utilizara el protocolo
                                       de encripcion 3des */

peer(172.16.1.1):                   /*Conexión con enrutador remoto*/
      passive: off                  /*Habilitado modo principal*/
      verify_identifier: on        /* Se verifica la autenticidad del
                                       otro enrutador*/
      hash_algorithm[0]: sha1       /* Se utilizara el protocolo
                                       de autenticación sha1 */
      encryption_algorithm[0]: 3des /* Se utilizara el protocolo
                                        de encripcion 3des */
      authentication_method[0]: pre_shared_key  /*método de autenticacion
                                                  clave pre compartida */
      my_identifier: address 192.168.1.1  /*verificación de autenticidad
                                            de enrutador local */
      peers_identifier: address 172.16.1.1 /*verificación de autenticidad
                                            de enrutador remoto */
      exchange_mode: main          /*método de intercambio de información
                                     principal */

connection(192-100):                /*Generacion de conexion*/
      src_range: 192.168.1.0/24     /*Red local en la que se va a
                                       encriptar la información */
      dst_range: 100.100.1.0/24    /*Red remota con la que se va a
                                       encriptar la información */
      src_ip: 192.168.1.1          /*Direccion del enrutador local*/
      dst_ip: 100.100.1.1          /*Direccion del enrutador remota*/
      admin_status: yes             /*Conexión Administrada*/
      authentication_algorithm: hmac_sha1 /* Se utilizara el protocolo
                                          de autenticación hmac_sha1 */
      encryption_algorithm: 3des      /* Se utilizara el protocolo
                                         de encripcion 3des */


peer(100.100.1.1):                    /*Conexión con enrutador remoto*/
      passive: off                    /*Habilitado modo principal*/
      verify_identifier: on           /* Se verifica la autenticidad del
                                      otro enrutador*/
      hash_algorithm[0]: sha1         /* Se utilizara el protocolo
                                      de autenticación sha1 */
      encryption_algorithm[0]: 3des   /* Se utilizara el protocolo
                                       de encripcion 3des */
      authentication_method[0]: pre_shared_key  /*método de autenticacion
                                                  clave pre compartida */
      my_identifier: address 192.168.1.1   /*verificación de autenticidad
                                             de enrutador local */
      peers_identifier: address 100.100.1.1 /*verificación de
                                      autenticidad de enrutador remoto */
      exchange_mode: main  /*método de intercambio de información
                                     principal */
```

3.3.4.6 **Archivos de Configuración IPv6**

Configuración RTRLNX 1

```
# IPSec IPv6 RTRLNX_1 192-2200
global:
      log: notify

connection(2200-2100):       /*Generacion de conexion Ipv6*/
      src_range: 2200:db8:3c4d:1::/64       /*Red local en la que se va a
                                             encriptar la información */
      dst_range: 2100:db8:3c4d:1::/64     /*Red remota con la que se va a
                                             encriptar la información */
      src_ip: 2200:db8:3c4d:1::1        /*Direccion del enrutador local*/
      dst_ip: 2100:db8:3c4d:1::1       /*Direccion del enrutador remoto*/
      admin_status: yes             /*Conexión Administrada*/
      authentication_algorithm: hmac_sha1  /* Se utilizara el protocolo
                                           de autenticación hmac_sha1 */
      encryption_algorithm: 3des    /* Se utilizara el protocolo
                                       de encripcion 3des */

peer(2100:db8:3c4d:1::1):
      passive: off                  /*Habilitado modo principal*/
      verify_identifier: on        /* Se verifica la autenticidad del
                                       otro enrutador*/
      hash_algorithm[0]: sha1       /* Se utilizara el protocolo
                                       de autenticación sha1 */
      encryption_algorithm[0]: 3des /* Se utilizara el protocolo
                                        de encripcion 3des */
      authentication_method[0]: pre_shared_key  /*método de autenticacion
                                                  clave pre compartida */
      my_identifier: address 2200:db8:3c4d:1::1  /*verificación de
                                        autenticidad de enrutador local */
      peers_identifier: address 2100:db8:3c4d:1::1 /*verificación de
                                       autenticidad de enrutador remoto */
      exchange_mode: main          /*método de intercambio de información
                                     principal */


connection(2200-2000):            /*Generacion de conexion Ipv6*/
      src_range: 2200:db8:3c4d:1::/64  /*Red local en la que se va a
                                          encriptar la información */
      dst_range: 2000:db8:3c4d:1::/64  /*Red remota con la que se va a
                                         encriptar la información */
      src_ip: 2200:db8:3c4d:1::1   /*Direccion del enrutador local*/
      dst_ip: 2000:db8:3c4d:1::1  /*Direccion del enrutador remoto*/
      admin_status: yes            /*Conexión Administrada*/
      authentication_algorithm: hmac_sha1  /* Se utilizara el protocolo
                                           de autenticación hmac_sha1 */
      encryption_algorithm: 3des    /* Se utilizara el protocolo
                                       de encripcion 3des */

peer(2000:db8:3c4d:1::1):
      passive: off                  /*Habilitado modo principal*/
      verify_identifier: on        /* Se verifica la autenticidad del
                                       otro enrutador*/
      hash_algorithm[0]: sha1       /* Se utilizara el protocolo
                                       de autenticación sha1 */
```

```
encryption_algorithm[0]: 3des /* Se utilizara el protocolo
                                 de encripcion 3des */
authentication_method[0]: pre_shared_key  /*método de autenticacion
                                            clave pre compartida */
my_identifier: address 2200:db8:3c4d:1::1 /*verificación de
                                 autenticidad de enrutador local */
peers_identifier: address 2000:db8:3c4d:1::1 /*verificación de
                                autenticidad de enrutador remoto */
exchange_mode: main          /*método de intercambio de información
                                principal */
```

3.4 PRUEBAS DE DESEMPEÑO

3.4.1 ESCENARIO 1: DIRECCIONAMIENTO IP VERSIÓN 4

En la figura 3.4 se muestra el diagrama de red utilizado como ambiente de pruebas para el funcionamiento del prototipo implementado.

El mismo que consta de:

RTRLNX_1: El cual hace las funciones de enrutador principal, éste equipo está implementado sobre un PC con sistema operativo LINUX DEBIAN, cuenta con tres interfaces de red:

- La ethernet 0 para conectarse a la red C 192.168.1.0, ésta es la red local.
- La ethernet 1 para conectarse a la red B 172.16.1.0.
- La ethernet 2 para conectarse a la red A 100.100.1.0.

RTRLNX_2: El cual hace las funciones de enrutador de borde, éste equipo está implementado sobre un PC con sistema operativo LINUX DEBIAN, cuenta con dos interfaces de red:

- La ethernet 0 directamente enlazada con la ethernet 2 del RNRLNX_1.
- La ethernet 1 para conectarse a la red A 100.100.1.0, ésta es la red local.

RTRLNX_3: El cual hace las funciones de enrutador de borde, éste equipo está implementado sobre una portátil en la cual está instalado VMWARE que nos permite tener una máquina virtual con sistema operativo LINUX DEBIAN. El RTRLNX_3 cuenta con dos interfaces de red:

- La ethernet 0 directamente enlazada con la ethernet 1 del RNRLNX_1.
- La ethernet 1 para conectarse a la red B 172.16.1.0, ésta es la red local.

CLIENTE_1: El cual cumple la función de cliente de la RED C, es un equipo portátil con dirección IP: 192.168.1.2 y gateway la ethernet 0 del RTRLNX_1.

CLIENTE_2: El cual cumple la función de cliente de la RED A, es un equipo portátil con dirección IP: 100.100.1.2 y gateway la ethernet 1 del RTRLNX_2.

CLIENTE_3: El cual cumple la función de cliente de la RED B, es una equipo portátil con dirección IP: 172.16.1.2 y gateway la ethernet 1 del RTRLNX_3.

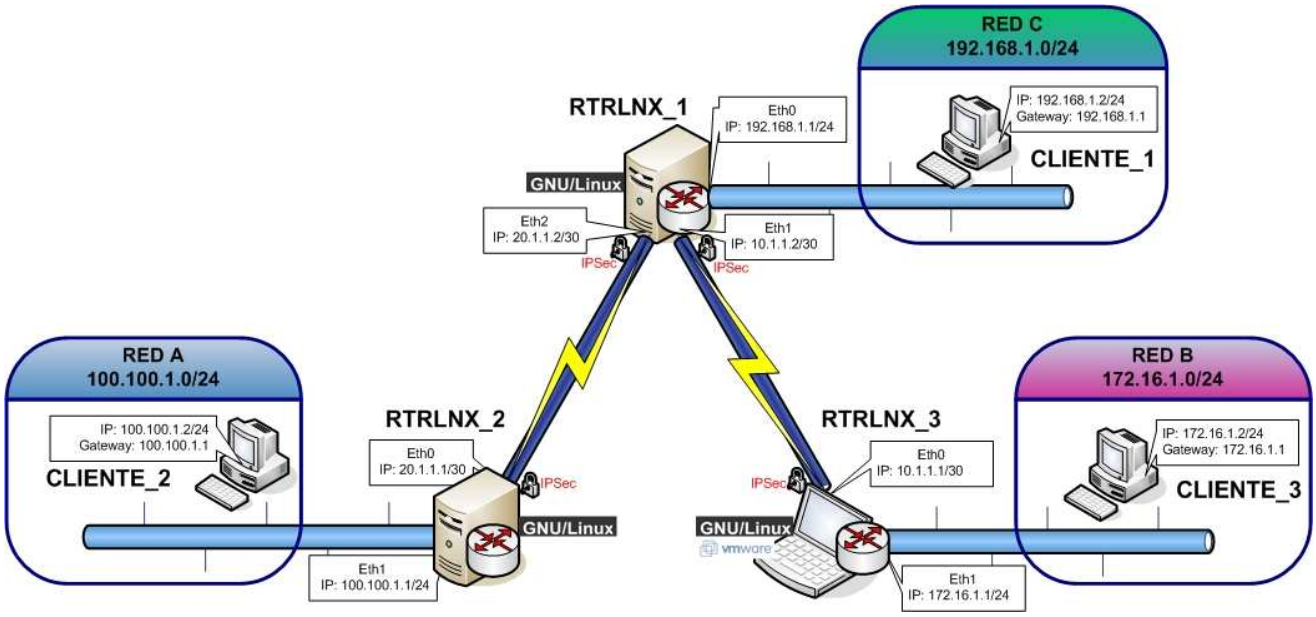

Figura 3- 4 Escenario de Pruebas IPv4

RED A	
Dirección de Red:	100.100.1.0
Sub Mascara:	255.255.255.0
Ancho de Banda UP	10 Mbps
Ancho de Banda DOWN	10 Mbps
ENRUTADOR RED A	
Dirección de Interface 0:	20.1.1.1
Sub Mascara:	255.255.255.252
Dirección de Interface 1:	100.100.1.1
Sub Mascara:	255.255.255.0
CLIENTE RED A	
Dirección de Interface:	100.100.1.2
Sub Mascara:	255.255.255.0
Gateway	100.100.100.1

RED B	
Dirección de Red:	172.16.1.0
Sub Mascara:	255.255.255.0
Ancho de Banda UP	40 Mbps
Ancho de Banda DOWN	40 Mbps
ENRUTADOR RED B	
Dirección de Interface 0:	10.1.1.1
Sub Mascara:	255.255.255.252
Dirección de Interface 1:	172.16.1.1
Sub Mascara:	255.255.255.0
CLIENTE RED B	
Dirección de Interface:	172.16.1.2
Sub Mascara:	255.255.255.0
Gateway	172.16.1.1

RED C	
Dirección de Red:	192.168.1.0
Sub Mascara:	255.255.255.0
Ancho de Banda UP	80 Mbps
Ancho de Banda DOWN	80 Mbps
ENRUTADOR RED C	
Dirección de Interface 0:	192.168.1.1
Sub Mascara:	255.255.255.0
Dirección de Interface 1:	10.1.1.2
Sub Mascara:	255.255.255.252
Dirección de Interface 2:	20.1.1.2
Sub Mascara:	255.255.255.252
CLIENTE RED C	
Dirección de Interface:	192.168.1.2
Sub Mascara:	255.255.255.0
Gateway	192.168.1.1

Figura 3- 5 Direcciones IPv4 asignadas a cada red

Para verificar la conectividad de red, y el correcto funcionamiento de los enrutadores implementados se utilizó los comandos ping y traceroute, además se utilizó un analizador de protocolos Wireshark, que nos permite visualizar los paquetes a través de la red.

Ping.- Comprueba el estado de la conexión con equipos remotos, por medio de paquetes de solicitud de eco y respuesta de eco.

Con cada protocolo implementado se realiza pruebas de conectividad ping.

Traceroute.- Permite seguir la pista de los paquetes que van desde un host a otro. Además de utilizarse para probar la conectividad, tracert puede utilizarse para verificar la latencia de red.

Wireshark.- Antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones. Permite ver todo el tráfico que pasa a través de una red, su funcionamiento es igual a la de tcpdump[42], pero añade una interfaz gráfica.

Wireshark es software libre, y se ejecuta sobre la mayoría de sistemas operativos Unix y compatibles, incluyendo Linux, Solaris, FreeBSD, NetBSD, OpenBSD, y Mac OS X, así como en Microsoft Windows.

3.4.1.1 **Pruebas Enrutamiento Estático**

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento estático.
- Usar el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_3.
- Usar el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_1.
- Utilizar el comando tracert para conocer la información de la ruta.

[42] tcpdump, es una herramienta en línea de comandos que permite analizar el trafico de la red. Permite capturar los paquetes transmitidos y recibidos de la red a la cual pertenece.

3.4.1.1.1 Ping

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _3 digitando:

  ping 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _1 digitando:

  ping 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
64 bytes from 172.16.1.2: icmp_seq=1 ttl=61 time=15.6 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=61 time=2.85 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=61 time=1.45 ms
64 bytes from 172.16.1.2: icmp_seq=4 ttl=61 time=10.4 ms
64 bytes from 172.16.1.2: icmp_seq=5 ttl=61 time=1.14 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4013ms
rtt min/avg/max/mdev = 1.141/6.308/15.633/5.769 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=62 time=1.53 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=62 time=29.1 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=62 time=1.41 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=62 time=1.23 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=62 time=1.73 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 1.239/7.008/29.118/11.056 ms
```

Análisis de resultados:

Se envían cinco solicitudes de ping al destino y se recibe la siguiente información de respuesta:

- bytes: tamaño del paquete ICMP.

- tiempo: tiempo transcurrido entre la transmisión y la respuesta.
- TTL: valor TTL predeterminado del dispositivo destino, menos la cantidad de routers en la ruta. El valor TTL predeterminado de Linux está configurado en 64.

  Debido a que los datagramas viajaron a través de tres routers para acceder al CLIENTE_3, el valor TTL devuelto es 61.

  El valor TTL para el segundo ping realizado es 62 debido a que los datagramas viajaron a través de dos routers para acceder a la CLIENTE_1.

3.4.1.1.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

  traceroute 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

  traceroute 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
traceroute to 172.16.1.2 (172.16.1.2), 30 hops max, 40 byte packets
 1   (100.100.1.1)  0.536 ms  0.456 ms  0.291 ms
 2   (10.1.1.1)  7.781 ms  7.674 ms  7.555 ms
 3   (172.16.1.2)  7.448 ms  7.281 ms  7.159 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 40 byte packets
 1    (100.100.1.1)  4.662 ms  4.433 ms  4.304 ms
 2    (20.1.1.2)  8.035 ms  7.923 ms  7.805 ms
 3    (192.168.1.2)  8.942 ms  8.935 ms  8.633 ms
```

Análisis de resultados:

El comando tracert permite ver la pérdida de conectividad con algún dispositivo final, y en general determinar la ruta que realiza el paquete para llegar a su destino.

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 10.1.1.1, (dirección IP de la ethernet 0 del RTRLNX_3)

  Y finalmente la dirección IP 172.16.1.2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 20.1.1.2, (dirección IP de la ethernet 2 del RTRLNX_1).

  Y finalmente la dirección IP 192.168.1.2, (dirección IP del CLIENTE_1).

3.4.1.1.3 Sniffer

Se envió un ping para verificar la conexión con el equipo remoto, desde el CLIENTE_2 de dirección IP: 100.100.1.2 se envía un paquete de solicitud eco al CLIENTE_3 de dirección IP: 172.16.1.2, 2 y se recibe un paquete de respuesta eco por parte de éste.

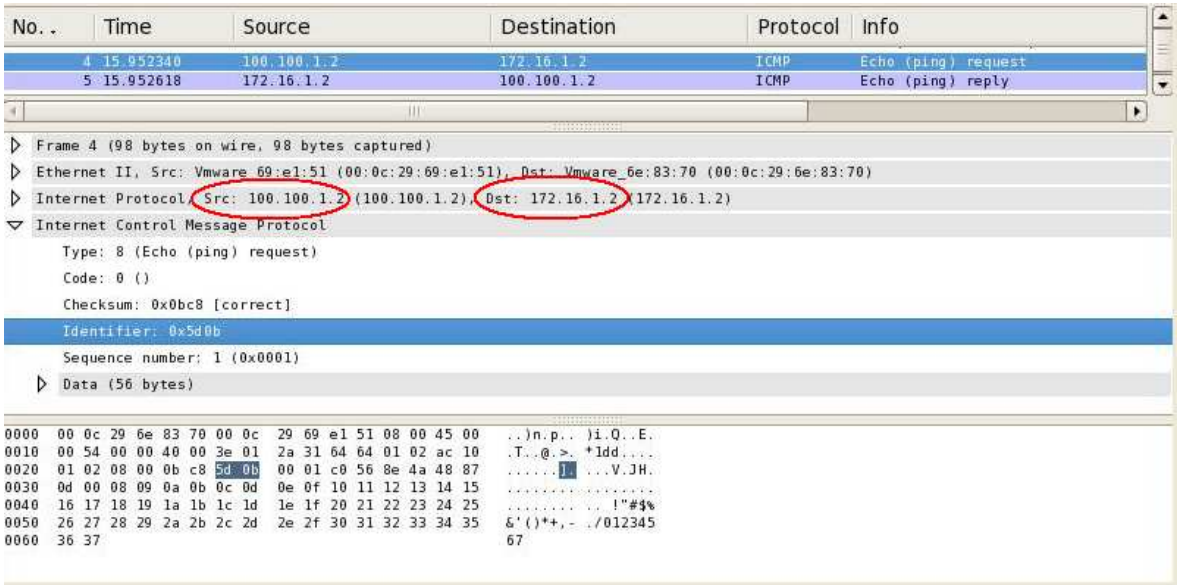

Figura 3- 6 Pruebas de enrutamiento estático IPv4, paquete REQUEST de ICMP

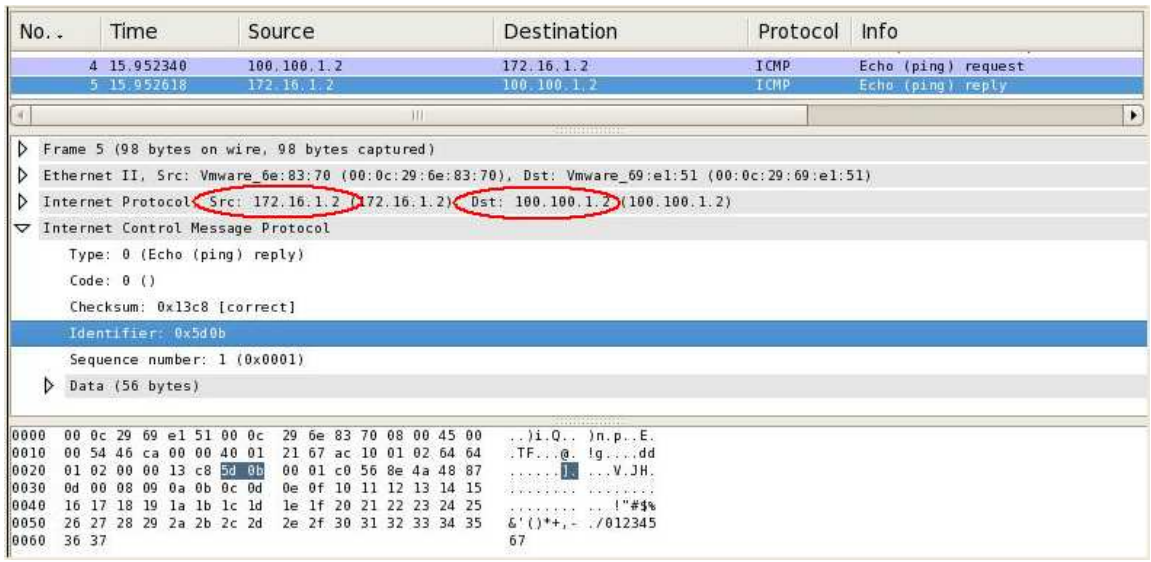

Figura 3- 7 Pruebas de enrutamiento estático IPv4, paquete REPLY de ICMP

Análisis de Resultados

En la tabla 3-16 se observa cómo se establece la conexión hacia un equipo. El proceso comienza verificando si dicho equipo está disponible enviando un paquete ICMP tipo ECHO REQUEST desde CLIENTE_2 al CLIENTE_3 en espera de un paquete ICMP tipo ECHO REPLY.

Con Wireshark, se lograron capturar los valores en hexadecimal de la solicitud y respuestas ICMP Eco, en la tabla 3-16 se muestran estos valores y su correspondencia en binario.

Los mensajes ICMP son construidos en el nivel de capa de red. IP encapsula el mensaje ICMP apropiado con una nueva cabecera IP , y con el campo “protocolo =1”. El formato de ICMP define el campo Type que identifica que mensaje se envía. La dirección IP del dispositivo remoto y el dispositivo local están contenidos en el paquete IP de capa 3.

ECO REQUEST		**ECO REPLY**		
Valor Hex	Valor Binario	Valor Hex	Valor Binario	Campo
08	0000 1000	00	0000 0000	Type
00	0000 0000	00	0000 0000	Code (0)
0b	0000 1011	13	0001 0011	Checksum
c8	1100 1000	c8	1100 1000	(correct)
5d	0101 1101	5d	0101 1101	Identifier
0b	0000 1011	0b	0000 1011	0X5d0b
00	0000 0000	00	0000 0000	Sequence Number
01	0000 0001	01	0000 0001	0X0001
00	0000 0000	00	0000 0000	Data (32 Bytes)
00	0000 0000	00	0000 0000	
00	0000 0000	00	0000 0000	
00	0000 0000	00	0000 0000	
.	.	.	.	
.	.	.	.	
.	.	.	.	
.	.	.	.	
00	0000 0000	00	0000 0000	
00	0000 0000	00	0000 0000	
00	0000 0000	00	0000 0000	
00	0000 0000	00	0000 0000	

Tabla 3-16 Tabulación paquetes Solicitud y Respuesta ECO, paquete ICMP

3.4.1.2 **Protocolo RIP**

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento dinámico.
- Comprobar el correcto funcionamiento de RIP utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_3.
- Comprobar el correcto funcionamiento de RIP utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_1.
- Utilizar el comando tracert para conocer la información de la ruta.

3.4.1.2.1 Ping

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _3 digitando:

  ping 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _1 digitando:

  ping 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
64 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=2.22 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=1.95 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=2.07 ms
64 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=3.41 ms
64 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=3.44 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4022ms
rtt min/avg/max/mdev = 1.953/2.621/3.440/0.665 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=62 time=1.66 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=62 time=1.33 ms
```

```
64 bytes from 192.168.1.2: icmp_seq=3 ttl=62 time=1.82 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=62 time=1.17 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=62 time=1.30 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 1.175/1.461/1.828/0.245 ms
```

Análisis de resultados:

Se envian cinco solicitudes de ping al destino y se recibe cinco paquetes, no se registra pérdida de paquetes.

3.4.1.2.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

  traceroute 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

  traceroute 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
traceroute to 172.16.1.2 (172.16.1.2), 30 hops max, 40 byte packets
 1   (100.100.1.1)  3.424 ms  5.275 ms  5.346 ms
 2   (10.1.1.1)  13.154 ms  13.310 ms  13.284 ms
 3   (172.16.1.2)  17.882 ms  18.022 ms  18.116 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 40 byte packets
 1   (100.100.1.1)  4.690 ms  4.549 ms  4.428 ms
 2   (20.1.1.2)  4.283 ms  4.153 ms  4.035 ms
 3   (192.168.1.2)  11.571 ms  11.444 ms  11.294 ms
```

Análisis de resultados:

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 10.1.1.1, (dirección IP de la ethernet 0 del RTRLNX_3)

  Y finalmente la dirección IP 172.16.1.2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 20.1.1.2, (dirección IP de la ethernet 2 del RTRLNX_1).

  Y finalmente la dirección IP 192.168.1.2, (dirección IP del CLIENTE_1).

3.4.1.2.3 Sniffer

Se reinicio el enrutador principal RTRLNX_1, y con Wireshark se comprobó como el enrutador actualiza su tabla de rutas, los paquetes RIP son transmitidos usando datagramas UDP (User Datagram Protocol) a través del puerto 520.

Los paquetes que envía son Request - Petición y Response - Respuesta.

REQUEST

En la figura 3-8 se muestra el formato del paquete RIP, con el campo comando=request, la versión=RIPv2, el campo Routing Domain el cual no es usado, y el campo Direcciones IP en este caso no son especificadas ya que aun no actualiza su tabla de enrutamiento.

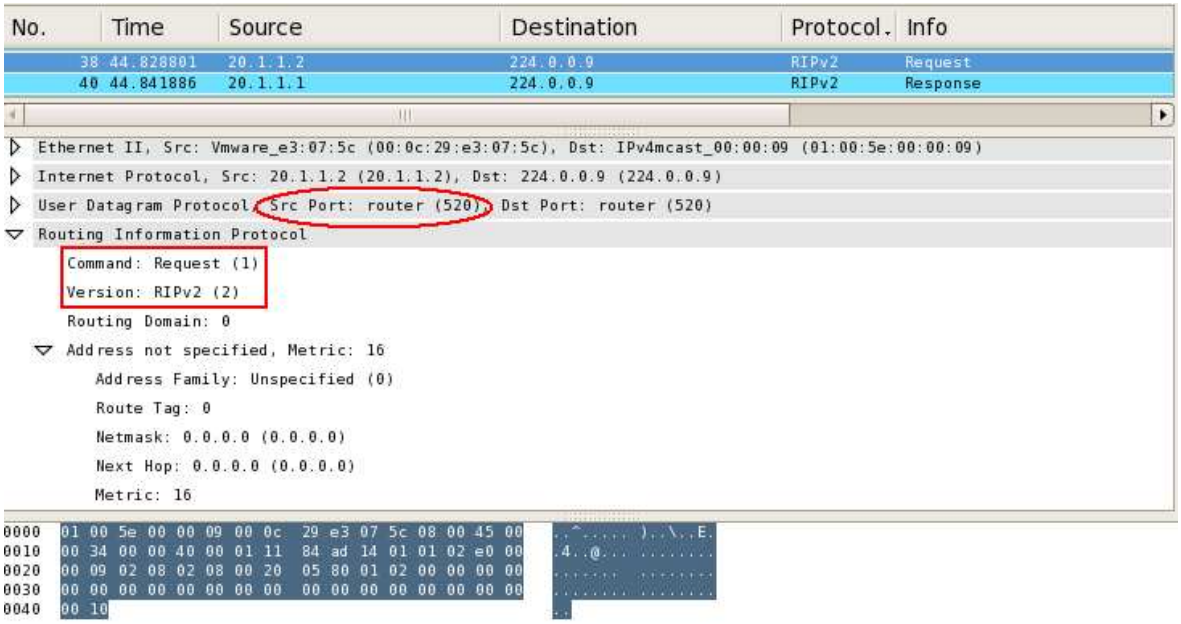

Figura 3- 8 Pruebas protocolo RIP, paquete REQUEST

RESPONSE

En la figura 3-9, se observa el campo comando=reponse, la versión=RIPv2, el campo Routing Domain el cual no es usado, y el campo Direcciones IP en este caso se actualizan todas las direcciones IP, en cada dirección IP se indica los campos familia de direcciones, la etiqueta de ruta, la dirección IP, la máscara de subred, el siguiente salto a ser alcanzado y la métrica definida para RIP.

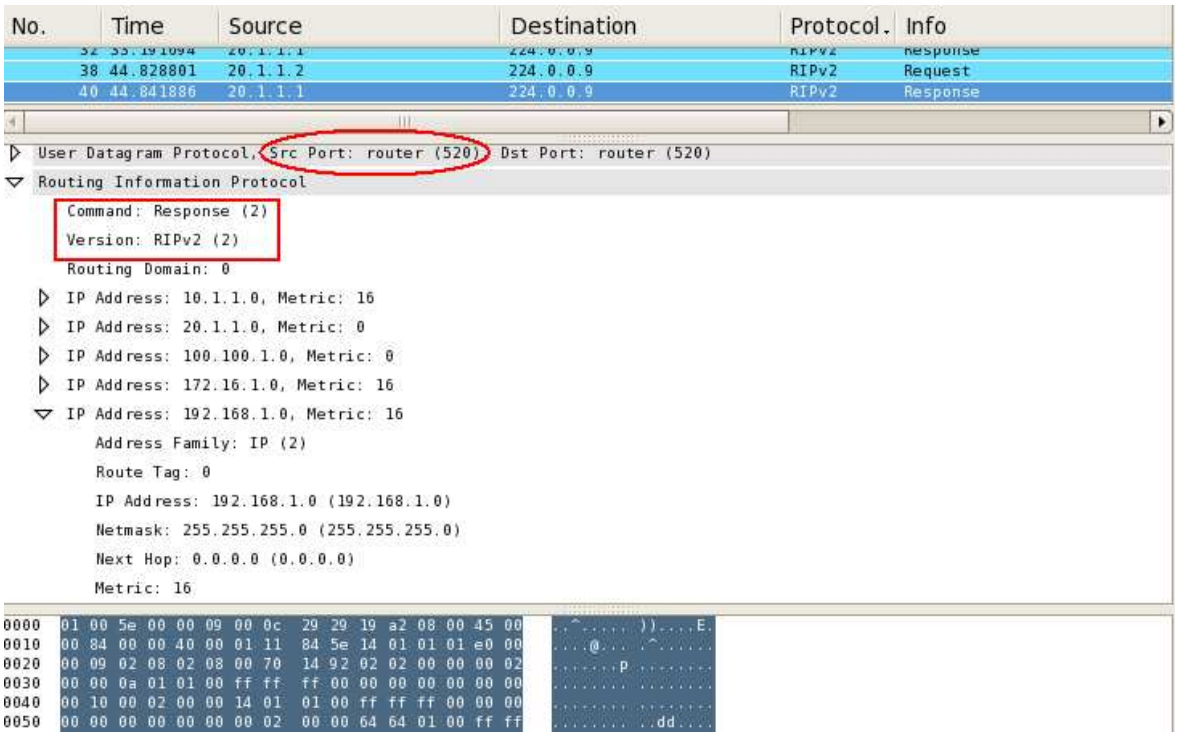

Figura 3- 9 Pruebas protocolo RIP, paquete RESPONSE

3.4.1.3 **Protocolo OSPF**

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento dinámico.
- Comprobar el correcto funcionamiento de OSPF utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_3.
- Comprobar el correcto funcionamiento de OSPF utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_1.
- Utilizar el comando tracert para conocer la información de la ruta.

3.4.1.3.1 Ping

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _3 digitando:

  ping 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _1 digitando:

  ping 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
64 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=2.06 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=1.70 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=2.06 ms
64 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=1.48 ms
64 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=2.06 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4017ms
rtt min/avg/max/mdev = 1.485/1.875/2.069/0.244 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=62 time=1.29 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=62 time=1.10 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=62 time=1.34 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=62 time=1.26 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=62 time=1.42 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4015ms
rtt min/avg/max/mdev = 1.102/1.286/1.425/0.109 ms
```

Análisis de resultados:

Se envian cinco solicitudes de ping al destino y se recibe cinco paquetes, no se registra pérdida de paquetes.

3.4.1.3.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

  traceroute 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

  traceroute 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
traceroute to 172.16.1.2 (172.16.1.2), 30 hops max, 40 byte packets
 1   (100.100.1.1)  0.386 ms  0.128 ms  0.217 ms
 2   (10.1.1.1)  8.183 ms  8.073 ms  7.955 ms
 3   (172.16.1.2)  10.396 ms  11.338 ms  11.225 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 40 byte packets
 1   (100.100.1.1)  0.443 ms  0.291 ms  0.141 ms
 2   (20.1.1.2)  6.232 ms  6.092 ms  5.975 ms
 3   (192.168.1.2)  7.917 ms  7.800 ms  7.602 ms
```

Análisis de resultados:

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 10.1.1.1, (dirección IP de la ethernet 0 del RTRLNX_3)

  Y finalmente la dirección IP 172.16.1.2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

El segundo salto es la IP 20.1.1.2, (dirección IP de la ethernet 2 del RTRLNX_1).

Y finalmente la dirección IP 192.168.1.2, (dirección IP del CLIENTE_1).

3.4.1.3.3 Sniffer

OSPF, permite a cada enrutador enviar mensajes del estado de las rutas a los routers vecinos. En la figura 3-10 se muestra el formato del paquete, compuesto por una cabecera con un campo versión= 2, que corresponde a OSPFv2, el campo tipo que indica el mensaje a ser enviado, en este caso corresponde al paquete Hello, el campo longitud de paquete = 48 Bytes. El campo source ospf router para identificar el origen del paquete, el campo chequeo de errores "checksum, el campo tipo de autenticación y el campo autenticación, a continuación se muestra el paquete HELLO, con cada uno de su campos descritos en el capítulo 1.

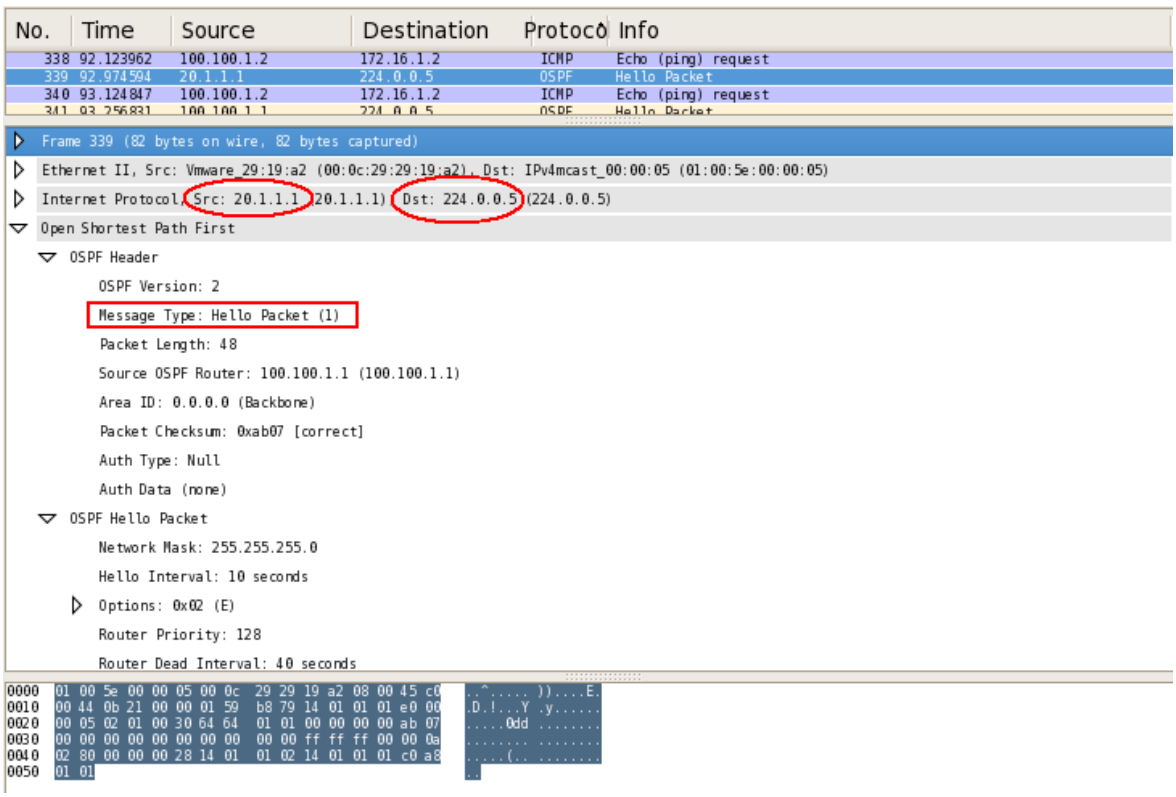

Figura 3- 10 Pruebas protocolo de enrutamiento OSPF

3.4.1.4 **Protocolo BGP**

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento dinámico.
- Comprobar el correcto funcionamiento de BGP utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_3.
- Comprobar el correcto funcionamiento de BGP utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_1.
- Utilizar el comando tracert para conocer la información de la ruta.

3.4.1.4.1 Ping

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _3 digitando:

  ping 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se verifica conectividad con el CLIENTE _1 digitando:

  ping 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
64 bytes from 172.16.1.2: icmp_seq=1 ttl=62 time=1.86 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=62 time=1.90 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=62 time=1.89 ms
64 bytes from 172.16.1.2: icmp_seq=4 ttl=62 time=2.64 ms
64 bytes from 172.16.1.2: icmp_seq=5 ttl=62 time=2.86 ms

--- 172.16.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4017ms
rtt min/avg/max/mdev = 1.860/2.232/2.863/0.432 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=62 time=1.29 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=62 time=1.10 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=62 time=1.34 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=62 time=1.26 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=62 time=1.42 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4015ms
rtt min/avg/max/mdev = 1.102/1.286/1.425/0.109 ms
```

Análisis de resultados:

Se envían cinco solicitudes de ping al destino y se recibe cinco paquetes, no se registra pérdida de paquetes.

3.4.1.4.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 100.100.1.2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

  traceroute 172.16.1.2

- Desde el CLIENTE_2 con IP: 100.100.1.2 se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

  traceroute 192.168.1.2

Resultados:

- Host 100.100.1.2 a 172.16.1.2

```
traceroute to 172.16.1.2 (172.16.1.2), 30 hops max, 40 byte packets
 1   (100.100.1.1)  2.844 ms  4.551 ms  4.436 ms
 2   (10.1.1.1)  7.625 ms  8.099 ms  9.630 ms
 3   (172.16.1.2)  12.533 ms  12.581 ms  12.604 ms
```

- Host 100.100.1.2 a 192.168.1.2

```
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 40 byte packets
 1   (100.100.1.1)  0.580 ms  0.388 ms  0.375 ms
 2   (20.1.1.2)  2.034 ms  2.054 ms  2.961 ms
 3   (192.168.1.2)  3.764 ms  4.430 ms  6.011 ms
```

Análisis de resultados:

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 10.1.1.1, (dirección IP de la ethernet 0 del RTRLNX_3)

  Y finalmente la dirección IP 172.16.1.2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 100.100.1.1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 20.1.1.2, (dirección IP de la ethernet 2 del RTRLNX_1).

  Y finalmente la dirección IP 192.168.1.2, (dirección IP del CLIENTE_1).

3.4.1.4.3 *Sniffer*

En la figura 3-11 se muestra como el puerto destino para bgp es el 179, con Wireshark además se puede comprobar como un paquete TCP puede ser enviado y como el protocolo de enrutamiento BGP permite la llegada del paquete a su destino.

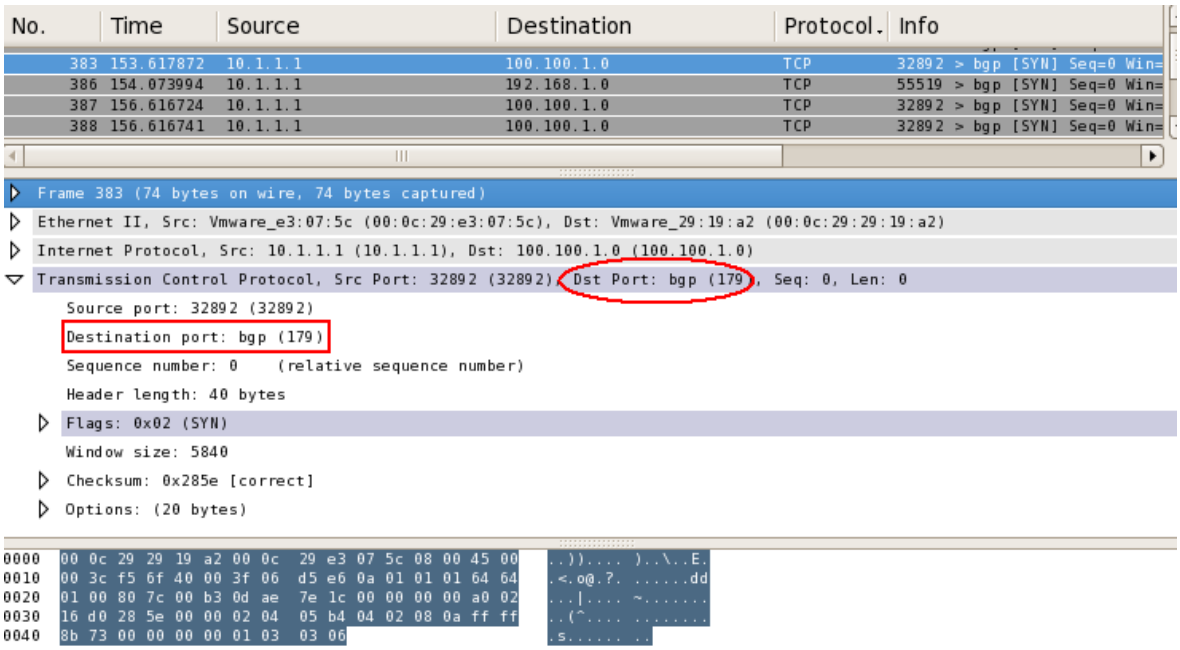

Figura 3- 11 Pruebas protocolo de enrutamiento BGP

3.4.1.5 ESP

En la figura 3-12 el datagrama IP es procesado y encapsulado dentro de ESP. El formato del paquete ESP nos indica el índice parámetro de seguridad (SPI) y el número de secuencia, la carga útil son los datos cifrados de protocolo IP.

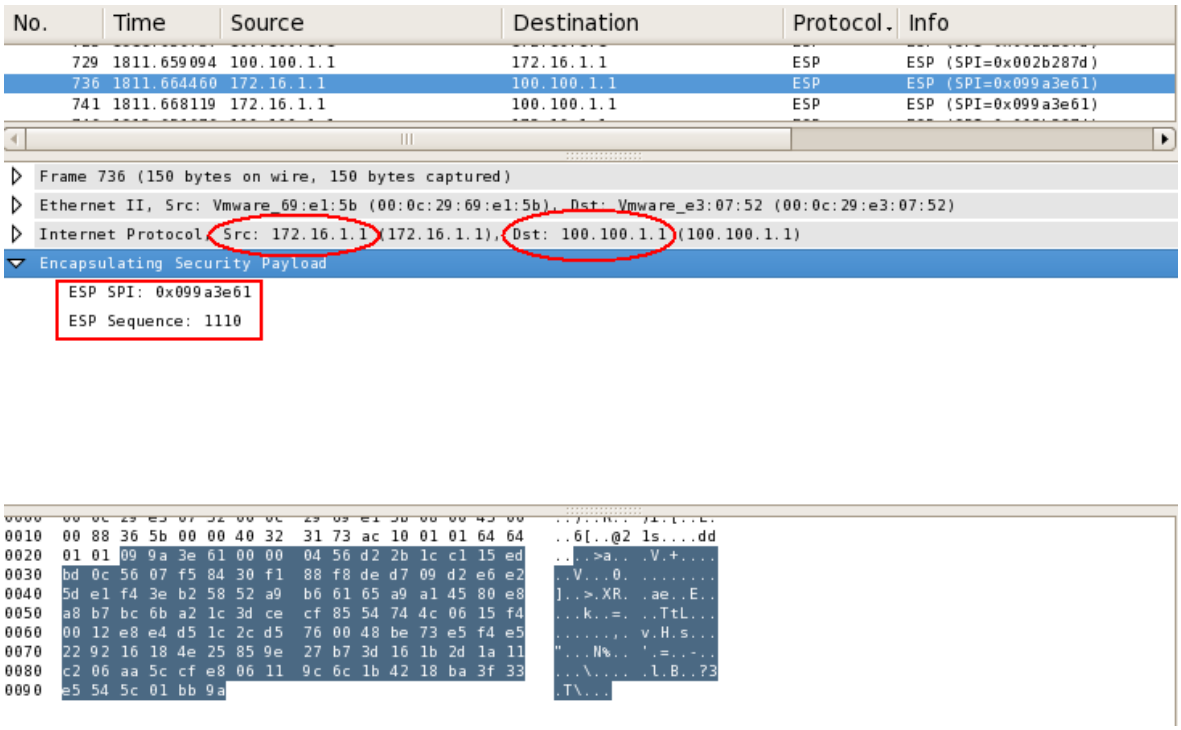

Figura 3- 12 Pruebas protocolo ESP

3.4.1.6 Calidad de Servicio, Medición de Anchos de Banda

Con el objetivo de probar el adecuado funcionamiento de las reglas de calidad de servicio se asigno anchos de banda específicos para cada red teniendo los siguientes valores:

Red	Enrutador	Ancho de Banda
A	RTRLNX_2	4 Mbps
B	RTRLNX_3	2 Mbps
C	RTRLNX_1	6 Mbps

Tabla 3-17 Anchos de banda asignados a las redes

Las mediciones se realizaron por medio del protocolo SNMP Protocolo de manejo simple de red - Simple Network Management Protocol y con la utilización del software PRTG Network Monitor[43] que nos permite receptar las tramas del protocolo que transmiten la información del ancho de banda utilizado en los intefaces del enrutador LINUX.

- RED A

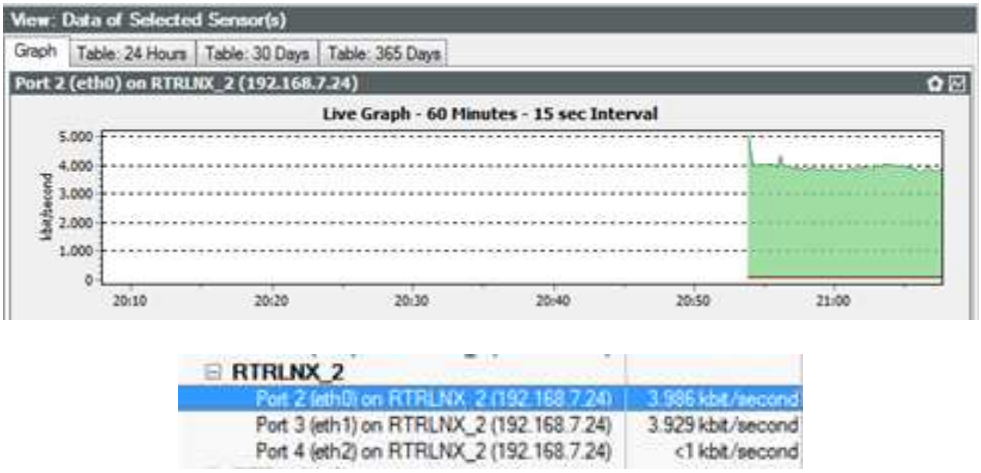

Figura 3- 13 Medición de ancho de banda Red A

- RED B

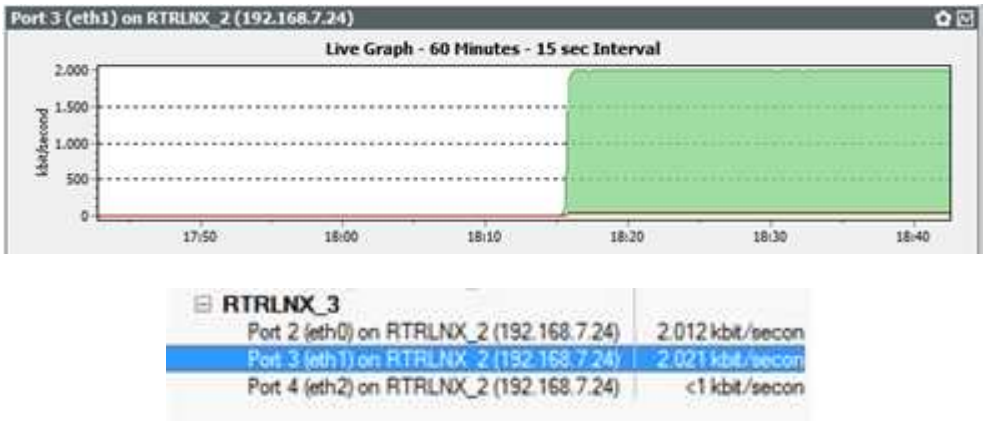

Figura 3- 14 Medición de ancho de banda Red B

[43] PRTG, es una aplicación de Windows que permite realizar el monitoreo y clasificación del uso del Ancho de Banda.

- RED C

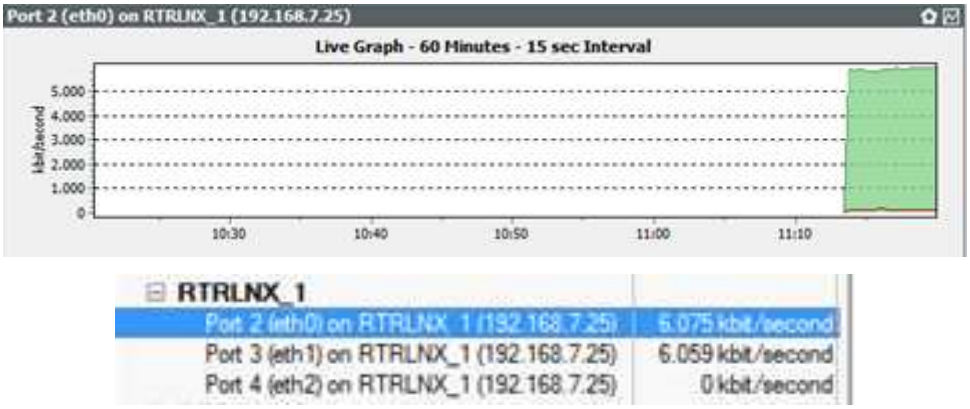

Figura 3- 15 Medición de ancho de banda Red C

Análisis de Resultados:

En base a las pruebas se observa que las reglas de calidad de servicio para asignación de anchos de banda funcionan adecuadamente y las asignaciones tiene un margen de variación de alrededor de +/- 1% lo que se considera aceptable.

3.4.2 ESCENARIO 2: DIRECCIONAMIENTO IP VERSIÓN 6

En la figura 3-16 se muestra el diagrama de la red utilizado como ambiente de pruebas para el funcionamiento del prototipo implementado con direccionamiento IPv6. El mismo que consta de:

RTRLNX_1: El cual hace las funciones de enrutador principal, este equipo está implementado sobre un PC con sistema operativo LINUX DEBIAN, cuenta con tres interfaces de red:

- La ethernet 0 para conectarse a la red C 2200:0db8:3c4d:0001:0000:0000:0000:0000, esta es la red local.
- La ethernet 1 para conectarse a la red B 2100:0db8:3c4d:0001:0000:0000:0000:0000.

- La Ethernet 2 para conectarse a la red A 2000:0db8:3c4d:0001:0000:0000:0000:0000.

RTRLNX_2: El cual hace las funciones de enrutador de borde, este equipo esta implementado sobre un PC con sistema operativo LINUX DEBIAN, cuenta con dos interfaces de red:

- La ethernet 0 directamente enlazada con la ethernet 2 del RNRLNX_1.
- La ethernet 1 para conectarse a la red A 2000:0db8:3c4d:0001:0000:0000:0000:0000, esta es la red local.

RTRLNX_3: El cual hace las funciones de enrutador de borde este equipo esta implementado sobre una portátil en la cual está instalado VMWARE que nos permite tener una máquina virtual con sistema operativo LINUX DEBIAN. El RTRLNX_3 cuenta con dos interfaces de red:

- La ethernet 0 directamente enlazada con la ethernet 1 del RNRLNX_1.
- La ethernet 1 para conectarse a la red B 2100:0db8:3c4d:0001:0000:0000:0000:0000, esta es la red local.

CLIENTE_1: El cual cumple la función de cliente de la RED C, es una portátil con dirección IP: 2200:0db8:3c4d:0001:0000:0000:0000:0002 y gateway la ethernet 0 del RTRLNX_1.

CLIENTE_2: El cual cumple la función de cliente de la RED A, es una portátil con dirección IP: 2000:0db8:3c4d:0001:0000:0000:0000:0002 y gateway la ethernet 1 del RTRLNX_2.

CLIENTE_3: El cual cumple la función de cliente de la RED B, es una portátil con dirección IP: 2100:0db8:3c4d:0001:0000:0000:0000:0002 y gateway la ethernet 1 del RTRLNX_3.

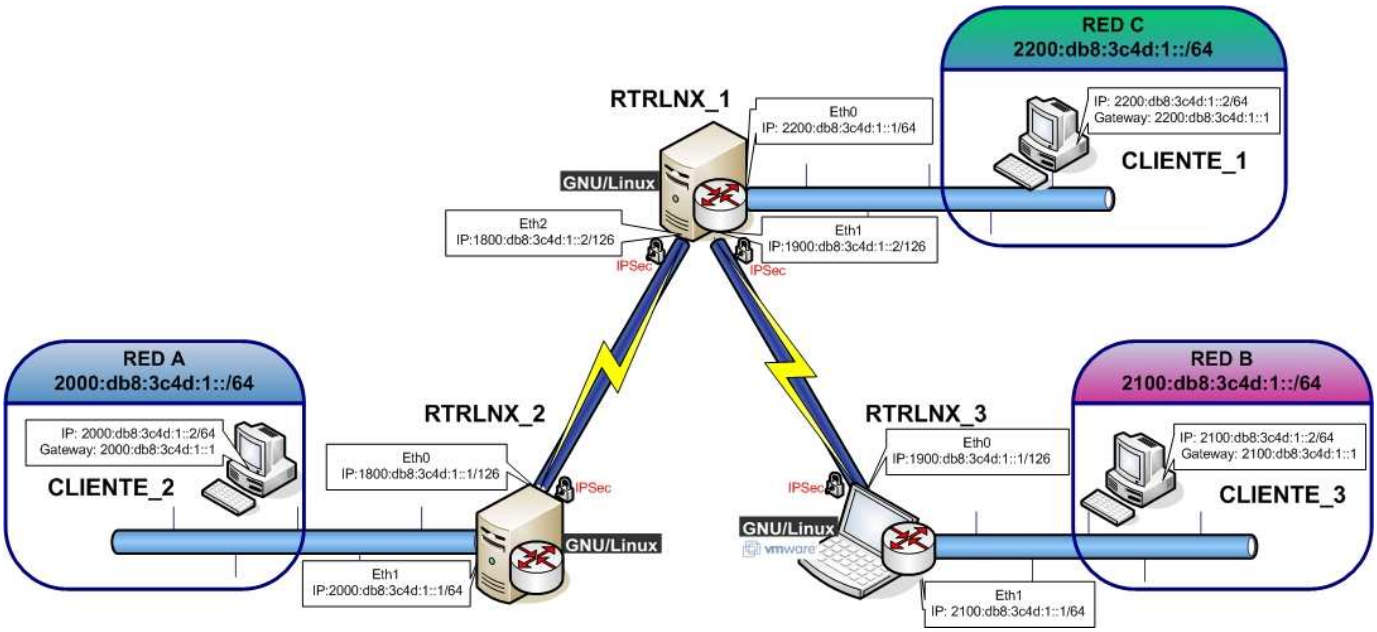

Figura 3- 16 Escenario de pruebas IPv6

RED A	
Dirección de Red:	2000:0db8:3c4d:0001:0000:0000:0000:0000
Longitud de Prefijo de Sub Red:	64
Ancho de Banda UP	10 Mbps
Ancho de Banda DOWN	10 Mbps
ENRUTADOR RED A	
Dirección de Interface 0:	1800:0db8:3c4d:0001:0000:0000:0000:0001
Longitud de Prefijo de Sub Red:	126
Dirección de Interface 1:	2000:0db8:3c4d:0001:0000:0000:0000:0001
Longitud de Prefijo de Sub Red:	64
CLIENTE RED A	
Dirección de Interface:	2000:0db8:3c4d:0001:0000:0000:0000:0002
Longitud de Prefijo de Sub Red:	64
Gateway:	2000:0db8:3c4d:0001:0000:0000:0000:0001

RED B	
Dirección de Red:	2100:0db8:3c4d:0001:0000:0000:0000:0000
Longitud de Prefijo de Sub Red:	64
Ancho de Banda UP	40 Mbps
Ancho de Banda DOWN	40 Mbps
ENRUTADOR RED B	
Dirección de Interface 0:	1900:0db8:3c4d:0001:0000:0000:0000:0001
Sub Mascara:	126
Dirección de Interface 1:	2100:0db8:3c4d:0001:0000:0000:0000:0001
Longitud de Prefijo de Sub Red:	64
CLIENTE RED B	
Dirección de Interface:	2100:0db8:3c4d:0001:0000:0000:0000:0002
Longitud de Prefijo de Sub Red:	64
Gateway:	2100:0db8:3c4d:0001:0000:0000:0000:0001

RED C	
Dirección de Red:	2200:0db8:3c4d:0001:0000:0000:0000:0000
Longitud de Prefijo de Sub Red:	64
Ancho de Banda UP	80 Mbps
Ancho de Banda DOWN	80 Mbps
ENRUTADOR RED C	
Dirección de Interface 0:	2200:0db8:3c4d:0001:0000:0000:0000:0001
Longitud de Prefijo de Sub Red:	64
Dirección de Interface 1:	1900:0db8:3c4d:0001:0000:0000:0000:0002
Longitud de Prefijo de Sub Red:	126
Dirección de Interface 2:	1800:0db8:3c4d:0001:0000:0000:0000:0002
Longitud de Prefijo de Sub Red:	126
CLIENTE RED C	
Dirección de Interface:	2200:0db8:3c4d:0001:0000:0000:0000:0002
Longitud de Prefijo de Sub Red:	64
Gateway:	2200:0db8:3c4d:0001:0000:0000:0000:0001

Figura 3- 17 Direcciones IPv6 asignadas a cada red

Para verificar la conectividad de red, bajo el direccionamiento IPv6 y el correcto funcionamiento de los enrutadores implementados se utilizaron los comandos pingv6 y tracert/traceroute, además se utilizó un analizador de protocolos Wireshark, que nos permite visualizar los protocolos a través de la red.

Pingv6.- Comprueba el estado de la conexión con equipos remotos, por medio de paquetes de solicitud de eco y respuesta de eco.

Con cada protocolo implementado se realiza pruebas de conectividad ping.

128 Echo Request (Solicitud de Eco).

129 Echo Reply (Respuesta de Eco).

3.4.2.1 **Enrutamiento Estático**

3.4.2.1.1 Ping

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento estático.
- Usar el comando pingv6 para enviar datagramas ICMPv6 desde el CLIENTE_2 al CLIENTE_3.
- Usar el comando pingv6 para enviar datagramas ICMPv6 desde el CLIENTE_2 al CLIENTE_1.

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _3 digitando:

ping 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _1 digitando:

ping 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
PING 2100:db8:3c4d:1::2(2100:db8:3c4d:1::2) 56 data bytes
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=7.53 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=5.46 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=3.58 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=3.48 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=2.79 ms

--- 2100:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4024ms
```

```
rtt min/avg/max/mdev = 2.797/4.574/7.538/1.727 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
PING 2200:db8:3c4d:1::2(2200:db8:3c4d:1::2) 56 data bytes
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=3.36 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=4.08 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=4.11 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=3.96 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=3.06 ms

--- 2200:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4019ms
rtt min/avg/max/mdev = 3.066/3.718/4.110/0.423 ms
```

Análisis de resultados:

Se envían cinco solicitudes de ping al destino y se recibe cinco paquetes, no se registra pérdida de paquetes.

3.4.2.1.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

  traceroute 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

  traceroute 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
traceroute to 2100:db8:3c4d:1::2 (2100:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  5.833 ms  5.693 ms  5.558 ms
 2   (1900:db8:3c4d:1::1)  19.743 ms  19.922 ms  20.027 ms
 3   (2100:db8:3c4d:1::2)  26.695 ms  26.878 ms  26.979 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
traceroute to 2200:db8:3c4d:1::2 (2200:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  73.056 ms  72.893 ms  72.740 ms
 2   (2200:db8:3c4d:1::1)  72.618 ms  72.495 ms  72.375 ms
 3   (2200:db8:3c4d:1::2)  129.935 ms  131.278 ms  131.631 ms
```

Análisis de resultados:

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 1900:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_3).

  Y finalmente la dirección IP 2100:db8:3c4d:1::2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 2200:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_1).

  Y finalmente la dirección IP 2200:db8:3c4d:1::2, (dirección IP del CLIENTE_1).

3.4.2.1.3 Sniffer

Se envió un ping para verificar la conexión con el equipo remoto, desde el CLIENTE_2 se envía un paquete de solicitud eco al CLIENTE_3 y se recibe un paquete de respuesta eco por parte de éste.

El Protocolo de Mensajes de Control de Internet Versión 6 (ICMPv6) es una nueva versión de ICMP y es una parte importante de la arquitectura IPv6. El protocolo

ICMPv6 es utilizado para detectar errores encontrados en la interpretación de paquetes y para el diagnóstico ICMPv6 ping.

Los paquetes ICMPv6 tienen el formato tipo, código y checksum.

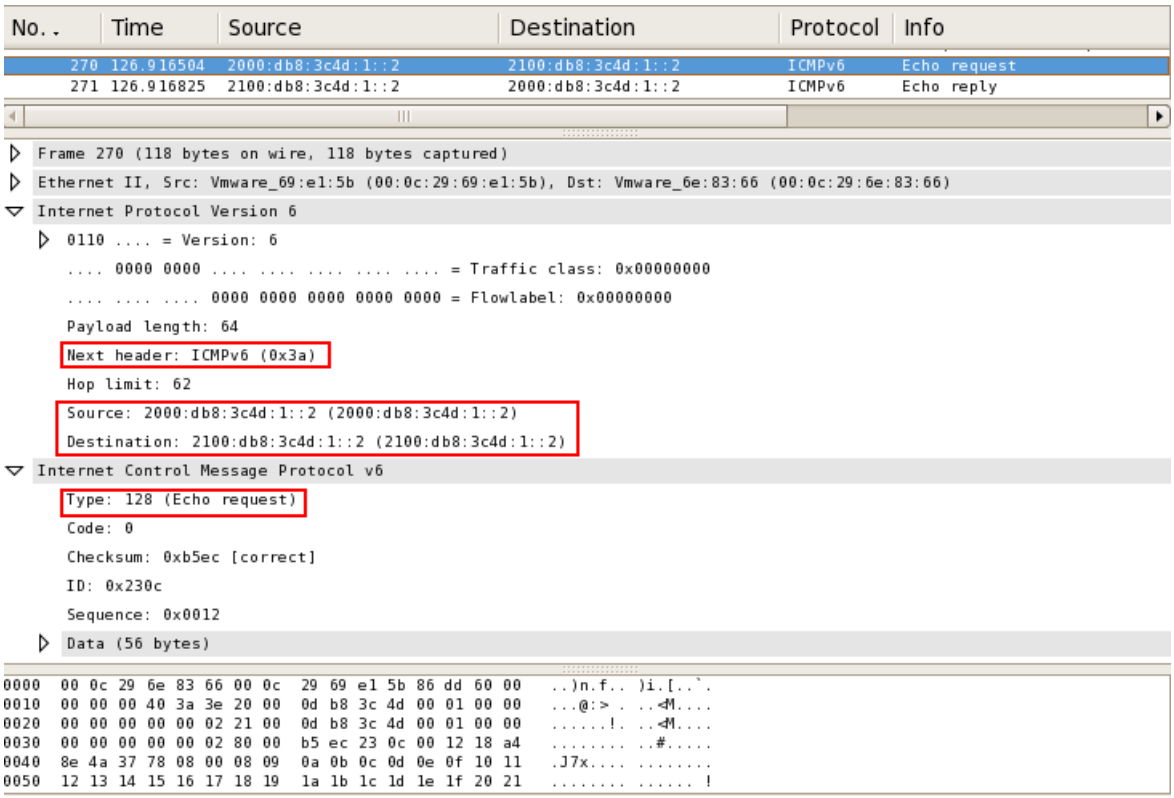

Figura 3- 18 Pruebas de enrutamiento estático IPv6, paquete REQUEST de ICMP

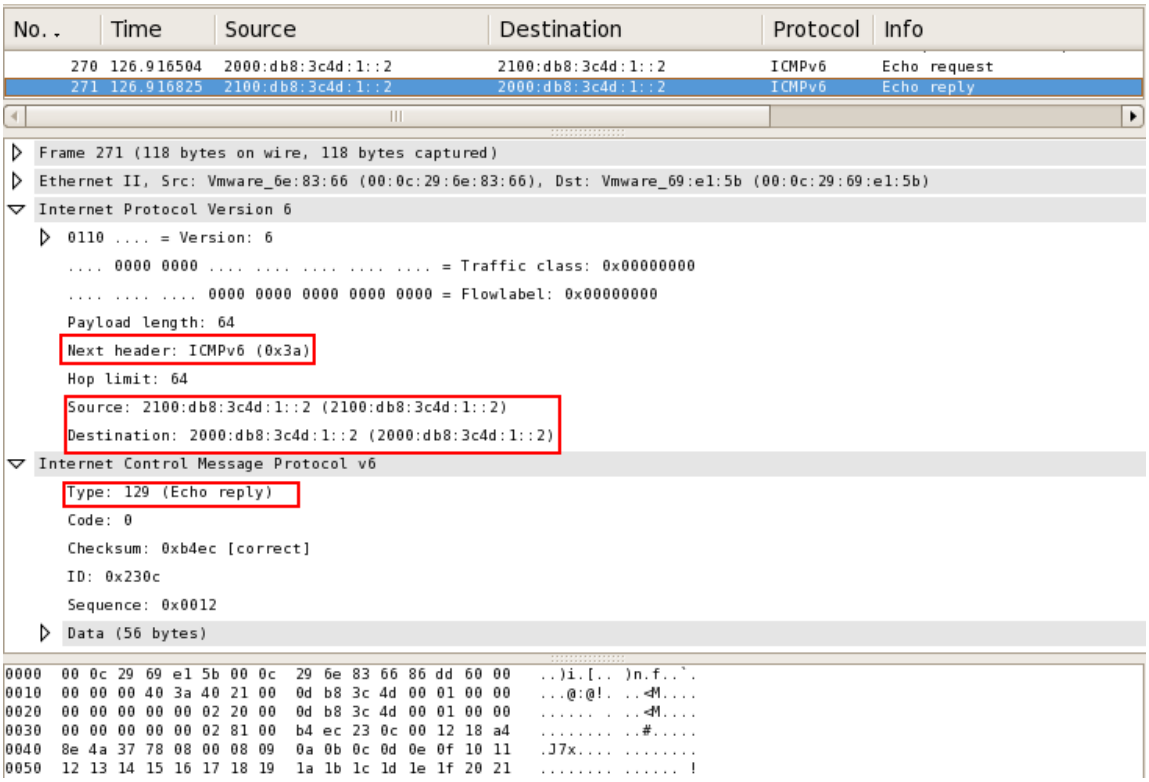

Figura 3- 19 Pruebas de enrutamiento estático IPv6, paquete REPLY de ICMP

Análisis de Resultados

En la figura 3-18 se observa cómo se establece la conexión hacia un equipo. El proceso comienza verificando si dicho equipo está disponible enviando un paquete ICMP tipo ECHO REQUEST desde CLIENTE_2 al CLIENTE_3 en espera de un paquete ICMP tipo ECHO REPLY.

Los mensajes ICMP son construidos en el nivel de capa de red. IPv6 encapsula el mensaje ICMP apropiado con una nueva cabecera IPv6 y con el campo “next header =ICMPv6”.

En la tabla 3-18 se indica el detalle del protocolo IPv6, su valor en hexadecimal y su correspondiente en binario.

PROTOCOLO DE INTERNET VERSION 6		
Valor Hex	Valor Binario	Campo
6	0110	Tipo (4 bits)

00	0000 0000	Clase de Tráfico (8 bits)
00	0000 0000	Etiqueta de Flujo (20 bits)
00	0000 0000	
0	0000	
00	0000 0000	Longitud de carga útil (16 bits)
64	0110 0100	
3a	0011 1010	Cabecera siguiente (8 bits)
62	0110 0010	Límite de saltos (8 bits)
20	0010 0000	Dirección origen (128 bits)
00	0000 0000	
0d	0000 1101	
b8	1011 1000	
3c	0011 1100	
4d	0100 1101	
00	0000 0000	
01	0000 0000	
00	0000 0000	
.	.	
.	.	
02	0000 0010	
21	0010 0001	Dirección destino (128 bits)
00	0000 0000	
0d	0000 1101	
b8	1011 1000	
3c	0011 1100	
4d	0100 1101	
00	0000 0000	
01	0000 0000	
00	0000 0000	
.	.	
.	.	
02	0000 0010	

Tabla 3-18 Tabulación paquete IPv6

El formato de ICMPv6 define:

El campo Tipo, que identifica que mensaje se envía.

Tipo= 128 ECO REQUEST

Tipo= 129 ECO REPLY

Identificador el cual es el mismo para el paquete ECO REQUEST y ECO REPLY.

ECO REQUEST		ECO REPLY		
Valor Hex	Valor Binario	**Valor Hex**	Valor Binario	Campo
128	10000001	**129**	10000001	Tipo
00	0000 0000	**0**	0000 0000	Código
b5	1011 0101	**b4**	1011 0100	Checksum (correcto)
ec	1110 1100	**ec**	1110 1100	
23	0010 0011	**23**	0010 0011	Identificador 0X230c
0c	0000 1100	**0c**	0000 1100	
00	0000 0000	**00**	0000 0000	Número de Secuencia 0X0012
12	0001 0010	**12**	0001 0010	
Datos		Datos		Datos (56 Bytes)

Tabla 3-19 Tabulación paquetes Solicitud y Respuesta ECO, ICMPv6

3.4.2.2 **Protocolo RIPng**

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento dinámico.
- Comprobar el correcto funcionamiento de RIPng utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_3.
- Comprobar el correcto funcionamiento de RIPng utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_1.
- Utilizar el comando tracert para conocer la información de la ruta.

3.4.2.2.1 Ping

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _3 digitando:

ping 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _1 digitando:

ping 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
PING 2100:db8:3c4d:1::2(2100:db8:3c4d:1::2) 56 data bytes
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=10.1 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=2.32 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=2.90 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=2.33 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=2.35 ms

--- 2100:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4018ms
rtt min/avg/max/mdev = 2.324/4.016/10.166/3.083 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
PING 2200:db8:3c4d:1::2(2200:db8:3c4d:1::2) 56 data bytes
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=13.7 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=2.58 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=1.85 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=1.74 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=2.23 ms

--- 2200:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 1.746/4.430/13.725/4.657 ms
```

Análisis de resultados:

Se envían cinco solicitudes de ping al destino y se recibe cinco paquetes, no se registra pérdida de paquetes.

3.4.2.2.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

  traceroute 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

  traceroute 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
traceroute to 2100:db8:3c4d:1::2 (2100:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  1.011 ms  0.464 ms  0.287 ms
 2   (1900:db8:3c4d:1::1)  14.466 ms  13.869 ms  13.758 ms
 3   (2100:db8:3c4d:1::2)  13.835 ms  13.887 ms  13.879 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
traceroute to 2200:db8:3c4d:1::2 (2200:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  0.705 ms  0.356 ms  7.580 ms
 2   (2200:db8:3c4d:1::1)  26.323 ms  26.019 ms  25.836 ms
 3   (2200:db8:3c4d:1::2)  33.379 ms  33.306 ms  33.146 ms
```

Análisis de resultados:

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 1900:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_3)

  Y finalmente la dirección IP 2100:db8:3c4d:1::2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 2200:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_1).

  Y finalmente la dirección IP 2200:db8:3c4d:1::2, (dirección IP del CLIENTE_1).

3.4.2.2.3 Sniffer

Con Wireshark se comprobó como el enrutador actualiza su tabla de rutas, los paquetes RIP son transmitidos usando datagramas UDP (User Datagram Protocol) a través del puerto 521.

Los paquetes que se envían son Request - Petición y Response - Respuesta.

REQUEST

En la figura 3-20 se muestra el formato del paquete RIPng:

Comando = request.

Versión = 1, que corresponde a RIPng.

Direcciones IP, en este caso no son especificadas ya que aún no actualiza su tabla de enrutamiento.

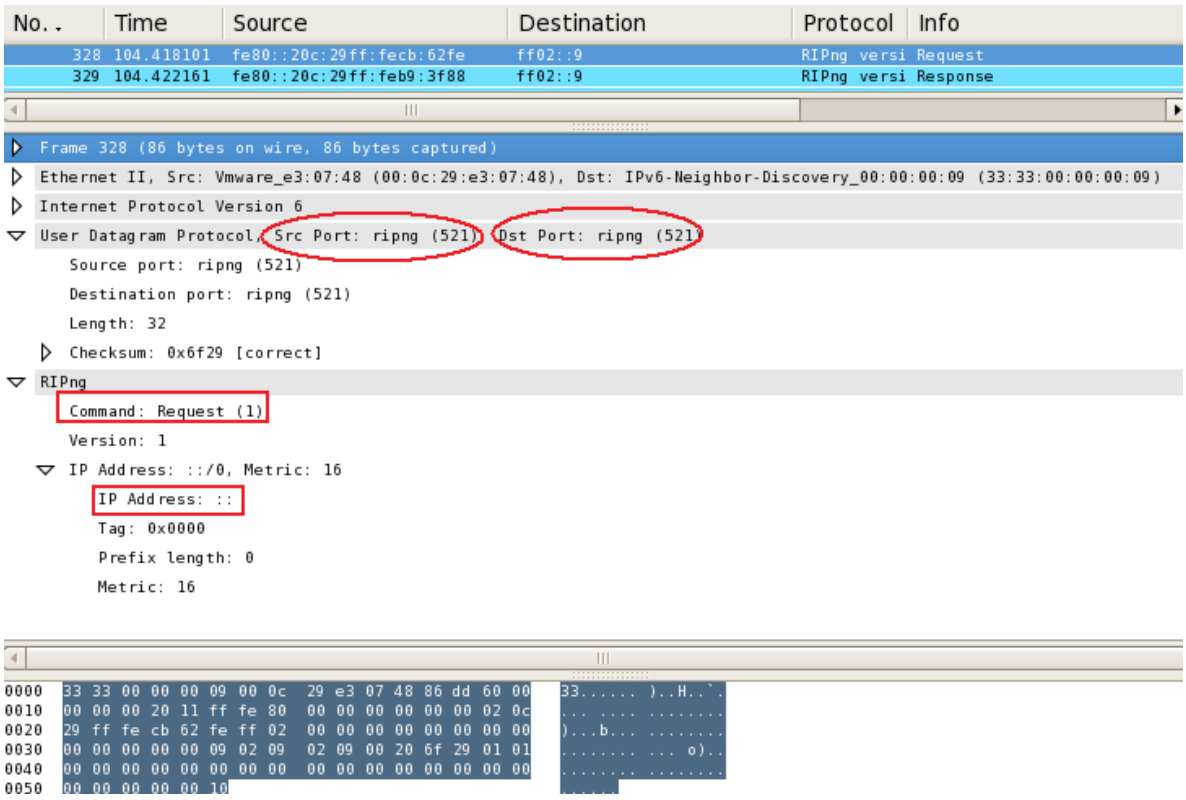

Figura 3- 20 Pruebas protocolo RIPng, paquete REQUEST

RESPONSE

En la figura 3-21 se muestra el formato del paquete RIPng:

Comando = response.

Versión = 1, que corresponde a RIPng.

Direcciones IPv6 en este caso se actualizan todas las direcciones IP.

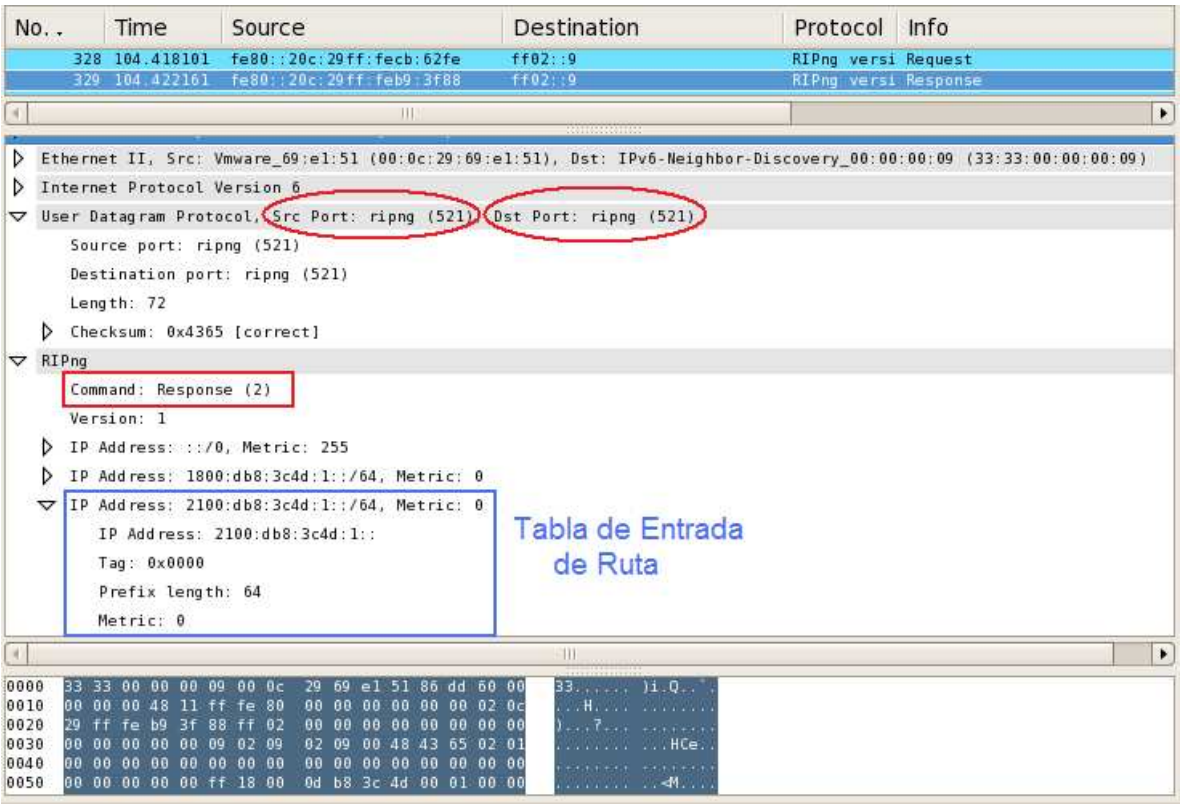

Figura 3- 21 Pruebas protocolo RIPng, paquete RESPONSE

Análisis de resultados:

En la tabla 3-20 se muestra los paquetes RIPng REQUEST y RESPONSE, su valor en hexadecimal y en binario.

Para el paquete REQUEST no se tiene entrada RTE (Tabla de entrada de ruta), debido a que no actualiza su tabla de enrutamiento.

Para el paquete RESPONSE tenemos ya especificadas las RTEs.

RIPng REQUEST		**RIPng RESPONSE**		
Valor Hex	Valor Binario	Valor Hex	Valor Binario	Campo
01	0000 0001	02	0000 0010	Comando (1 Byte)
01	0000 0001	01	0000 0001	Versión = 1 (1 Byte)
Dirección IPv6 = ::/0		Dirección IPv6 = 1800:db8:3c4d:1::/64		Entrada RTE1 (20 Byte)
		Dirección IPv6 = 2100:db8:3c4d:1::/64		Entrada RTE2 (20 Bytes)

Tabla 3-20 Tabulación paquetes solicitud y respuesta RIPng

En la tabla 3-21 se explica la RTE2, aquí se detalla que el prefijo IPv6 es la dirección IPv6 destino. La etiqueta de ruta en este caso no se usa, la longitud del prefijo que indica la parte más significativa de la dirección IPv6, y el valor de la métrica.

Tabla de entrada de Ruta 2		
Valor en Hex	Campo	
2100:db8:3c4d:1::	prefijo IPV6	(16 Bytes)
0000	etiqueta de Ruta	(2 Bytes)
64	longitud del prefijo	(1 Byte)
0	métrica	(1 Byte)

Tabla 3-21 Tabulación Tabla de entrada de Ruta 2

3.4.2.3 **Protocolo OSPF Versión 3**

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento dinámico.
- Comprobar el correcto funcionamiento de OSPFv3 utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_3.
- Comprobar el correcto funcionamiento de OSPFv3 utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_1.
- Utilizar el comando traceroute para conocer la información de la ruta

3.4.2.3.1 Ping

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _3 digitando:

ping 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _1 digitando:

ping 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
PING 2100:db8:3c4d:1::2(2100:db8:3c4d:1::2) 56 data bytes
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=2.81 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=2.21 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=3.10 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=3.34 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=2.33 ms

--- 2100:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4020ms
rtt min/avg/max/mdev = 2.213/2.762/3.343/0.440 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
PING 2200:db8:3c4d:1::2(2200:db8:3c4d:1::2) 56 data bytes
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=1.93 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=2.18 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=1.53 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=4.71 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=2.08 ms

--- 2200:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 1.538/2.491/4.713/1.132 ms
```

Análisis de resultados:

Se envían cinco solicitudes de ping al destino y se recibe cinco paquetes, no se registra pérdida de paquetes.

3.4.2.3.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

traceroute 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

traceroute 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
traceroute to 2100:db8:3c4d:1::2 (2100:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  1.578 ms  0.952 ms  0.447 ms
 2   (1900:db8:3c4d:1::1)  20.104 ms  19.993 ms  15.762 ms
 3   (2100:db8:3c4d:1::2)  29.040 ms  28.987 ms  28.865 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
traceroute to 2200:db8:3c4d:1::2 (2200:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  0.481 ms  0.265 ms  0.238 ms
 2   (2200:db8:3c4d:1::1)  13.324 ms  13.168 ms  21.319 ms
 3   (2200:db8:3c4d:1::2)  21.262 ms  21.507 ms  21.666 ms
```

Análisis de resultados:

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 1900:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_3)

  Y finalmente la dirección IP 2100:db8:3c4d:1::2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 2200:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_1).

Y finalmente la dirección IP 2200:db8:3c4d:1::2, (dirección IP del CLIENTE_1).

3.4.2.3.3 Sniffer

En la figura 3-31 se muestra el formato del paquete OSPF, compuesto por una cabecera, y los tipos de paquetes a ser enviados como son el Hello, LSA, LSU; DB, etc.

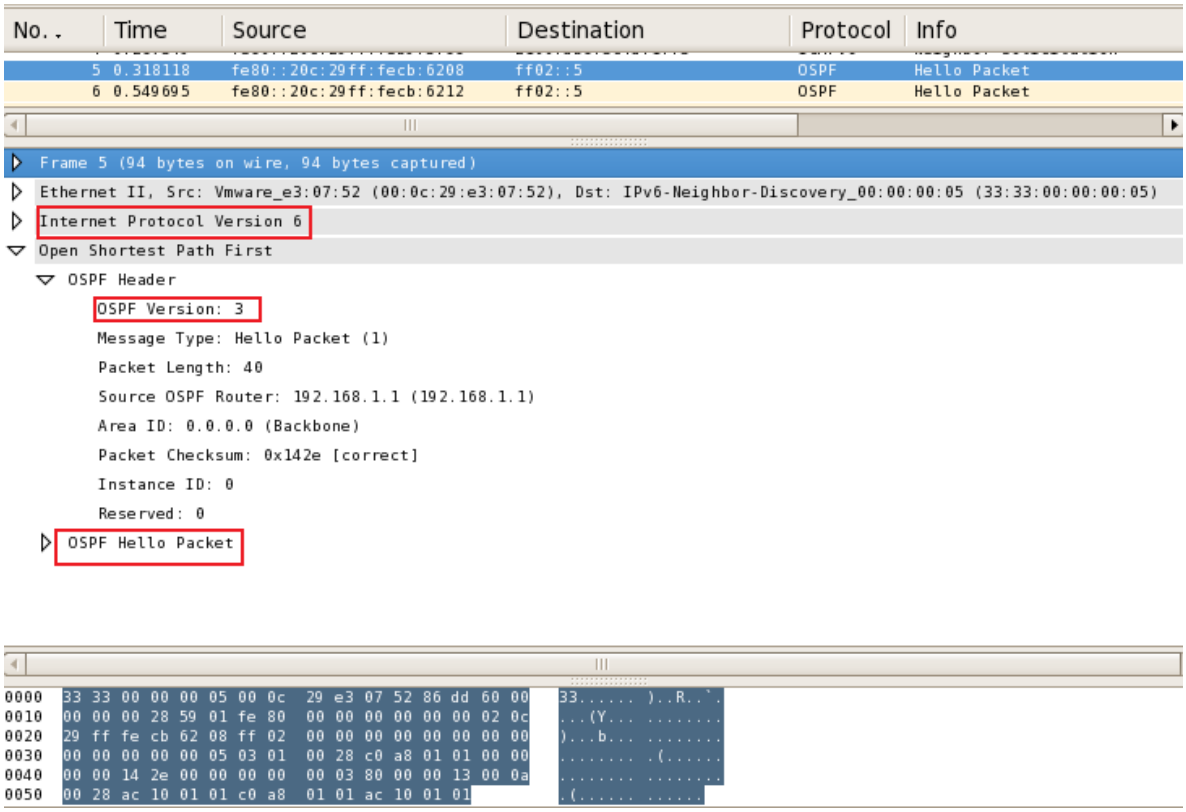

Figura 3- 22 Pruebas protocolo OSPFv3

Análisis de resultados:

La cabecera de OSPFv3 contiene los siguientes campos:

Versión= 3, que corresponde a OSPFv3.

Tipo de mensaje= Paquete HELLO.

Longitud de paquete = 40 Bytes.

Source ospf router para identificar el origen del paquete, el campo chequeo de errores “checksum” que nos muestra [correcto] lo que implica que no existen errores, los siguientes campos Instance ID y reserved con valores de 0 Bytes.

OSPF v3		
Valor	Valor Binario	Campo
03	0000 0011	Versión (1 Byte)
01	0000 0001	Tipo (1 Byte)
00	0000 0000	Longitud del Paquete (2 Bytes)
40	0100 0000	
Dirección: 192.168.1.1	11000000	ID del ruteador (4 Bytes)
	10101000	
	00000001	
	00000001	
Dirección: 0.0.0.0	00000000	ID del área (4 Bytes)
	00000000	
	00000000	
	00000000	
14	0001 0100	Checksum (2 Bytes) (correcto)
2e	0010 1110	
00	0000 0000	Instancia ID (1 Bytes)
00	0000 0000	Reservado (1 Bytes)

Tabla 3-22 Tabulación paquete OSPFv3

Paquete HELLO

Uno de los paquetes que se transmite dentro de OSPF es el paquete HELLO, cuyos campos más importantes son el DR Designated Router (Ruteador designado) y el BDR Backup Designated Router (Ruteador designado de respaldo), los cuales son utilizados por todos los ruteadores de una red para intercambiar información con éstos enrutadores designados.

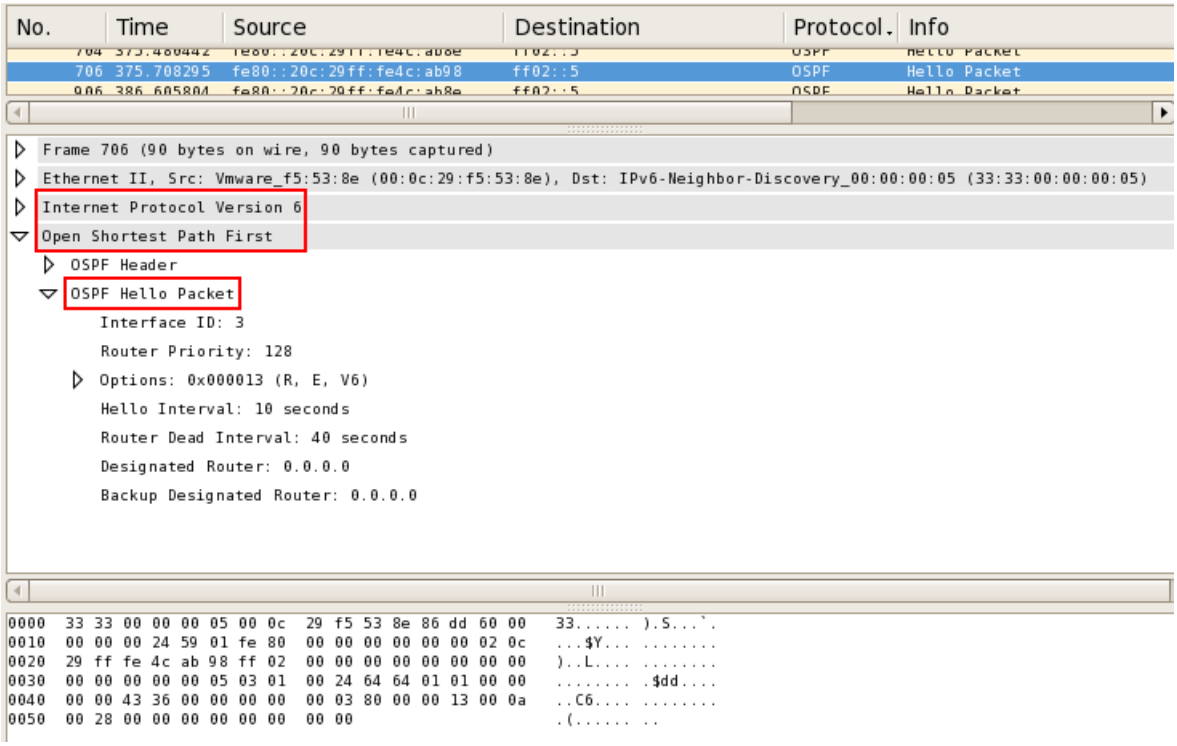

Figura 3- 23 Pruebas protocolo OSPF, paquete HELLO

3.4.2.4 **Protocolo BGP**

Objetivo:

- Comprobar el correcto funcionamiento de los ruteadores implementados bajo el esquema de enrutamiento dinámico.
- Comprobar el correcto funcionamiento de BGPv4 utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_3.
- Comprobar el correcto funcionamiento de BGPv4 utilizado el comando ping para enviar datagramas ICMP desde el CLIENTE_2 al CLIENTE_1.
- Utilizar el comando tracert para conocer la información de la ruta.

3.4.2.4.1 Ping

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _3 digitando:

ping 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2 se verifica conectividad con el CLIENTE _1 digitando:

ping 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
PING 2100:db8:3c4d:1::2(2100:db8:3c4d:1::2) 56 data bytes
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=4.26 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=3.14 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=2.98 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=2.39 ms
64 bytes from 2100:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=3.64 ms

--- 2100:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4024ms
rtt min/avg/max/mdev = 2.392/3.284/4.261/0.632 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
PING 2200:db8:3c4d:1::2(2200:db8:3c4d:1::2) 56 data bytes
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=1 ttl=62 time=2.34 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=2 ttl=62 time=2.04 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=3 ttl=62 time=3.43 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=4 ttl=62 time=1.47 ms
64 bytes from 2200:db8:3c4d:1::2: icmp_seq=5 ttl=62 time=1.59 ms

--- 2200:db8:3c4d:1::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 1.471/2.175/3.433/0.704 ms
```

Análisis de resultados:

Se envian cinco solicitudes de ping al destino y se recibe cinco paquetes, no se registra pérdida de paquetes.

3.4.2.4.2 Traceroute

Procedimiento:

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _3 digitando:

  traceroute 2100:db8:3c4d:1::2

- Desde el CLIENTE_2 con IP: 2000:db8:3c4d:1::2, se comprueba la información de la ruta hacia el CLIENTE _1 digitando:

  traceroute 2200:db8:3c4d:1::2

Resultados:

- Host 2000:db8:3c4d:1::2 a 2100:db8:3c4d:1::2

```
traceroute to 2100:db8:3c4d:1::2 (2100:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  0.694 ms  0.332 ms  0.196 ms
 2   (1900:db8:3c4d:1::1)  33.172 ms  33.167 ms  33.149 ms
 3   (2100:db8:3c4d:1::2)  33.923 ms  33.707 ms  33.580 ms
```

- Host 2000:db8:3c4d:1::2 a 2200:db8:3c4d:1::2

```
traceroute to 2200:db8:3c4d:1::2 (2200:db8:3c4d:1::2), 30 hops max, 40
byte packets
 1   (2000:db8:3c4d:1::1)  1.961 ms  0.873 ms  0.771 ms
 2   (2200:db8:3c4d:1::1)  1.431 ms  4.260 ms  1.552 ms
 3   (2200:db8:3c4d:1::2)  41.669 ms  41.556 ms  41.431 ms
```

Análisis de resultados:

- En el primer caso se verifica que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 1900:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_3).

  Y finalmente la dirección IP 2100:db8:3c4d:1::2, (dirección IP del CLIENTE_3).

- En el segundo traceroute realizado se visualiza que el primer salto lo realiza a la dirección IP 2000:db8:3c4d:1::1 (dirección IP de la ethernet 1 del RTRLNX_2) corresponde al gateway configurado en el CLIENTE_2.

  El segundo salto es la IP 2200:db8:3c4d:1::1, (dirección IP de la ethernet 0 del RTRLNX_1).

  Y finalmente la dirección IP 2200:db8:3c4d:1::2, (dirección IP del CLIENTE_1).

CAPÍTULO 4. ESTIMACIÓN DE COSTOS

4.1 INTRODUCCIÓN

El poder brindar una solución de bajo costo para pequeñas y medianas empresas es un objetivo a ser alcanzado en el presente proyecto de titulación, razón por la cual en este capítulo se realizará una estimación del costo del prototipo implementado.

Se efectuará una propuesta del precio del prototipo junto con el detalle de las características técnicas y económicas para su comercialización.

El prototipo al tener una plataforma estable como lo es el sistema operativo Linux, el contar con la eficiencia de su código fuente que implica mayor velocidad en las aplicaciones, y ser implementado bajo software libre, hace del mismo una alternativa muy confiable para ser distribuido y comercializado en pequeñas y medianas empresas.

Se concluirá el estudio con una comparación técnico – económica con ruteadores de marcas reconocidas en el mercado que presenten características similares a las del prototipo router dual ipv4/ipv6.

4.2 ANÁLISIS DE COSTOS DEL PROTOTIPO

Para estimar el costo final del prototipo, se deben tomar en cuenta 4 parámetros importantes:

- Costo del equipo.
- Costo de la implementación.
- Costo de la investigación.
- Costo de documentación.

4.2.1 COSTO DEL EQUIPO

El costo del equipo, es el menos representativo entre los parámetros a analizarse, ya que este era uno de los objetivos del proyecto, y teniendo en cuenta que el equipamiento se proyecta principalmente a la reutilización de equipos o la utilización de equipos ensamblados, se obtuvo como se deseaba un equipo de bajo valor.

Una ventaja del sistema operativo Linux, es que nos permite utilizar mecanismos de seguridad propios de equipos servidores en máquinas ensambladas o clones, lo que para un sistema de producción puede ser muy útil para asegurar el servicio, pero para el presente análisis no se toma en cuenta ninguna de estas medidas de seguridad y se analizará únicamente el costo de cada equipo ruteador.

El costo del software implementado es cero (0 USD) ya que se utilizó en su totalidad software libre con licencia GNU/GPL lo que permitió verificar la fortaleza del código abierto y fundamentalmente abaratar costos.

La tabla siguiente detalla el costo de adquisición del hardware:

Ítem	Cantidad	Característica	Precio Unitario (USD)	Precio Total (USD)
Procesador	1	**INTEL DUAL CORE E2200 2.2GHZ**	77	77
Memoria	2	**ADATA 1GB PC-800**	10	20
Tarjeta Madre	1	**INTEL DG31PR S775,1333GHZ,DDR2,V,S,R**	66	66
Disco Duro	1	**160GB SAMSUNG SATA 7200RPM**	42	42
Red	2	**Giga bit Ethernet D-LINK modelo DGE-530T**	30	60
Case	1	**CASE DLUXE DLC-MF453 MIDTOWER 24P**	39	39
TOTAL				**304**

Tabla 4-1 Detalle de costo de hardware del prototipo

4.2.2 COSTO DE IMPLEMENTACIÓN

Para el análisis del costo de la implementación se tomará como parámetro principal el tiempo que lleva la tarea de instalación y configuración y tomando el

hardware listo para la instalación del sistema operativo además teniendo en cuenta que el implementador posee la suficiente experiencia para modificar los archivos de configuración.

- **Detalle de duración por cada actividad:**

ACTIVIDAD	TIEMPO (horas)
Instalación de sistema operativo	1
Recopilación de kernel y parches	6
Instalación de software de ruteo (XORP)	1
Instalación de software de calidad de servicio (CBQ.INIT)	1
Instalación de software de Ipsec (RACOON)	1
Configuración de XORP	8
Configuración de CBQ.INIT	5
Configuración de FREESWAN	10
TOTAL	33

Tabla 4-2 Detalle de costo de implementación

4.2.3 COSTO DE INVESTIGACIÓN

Es el parámetro más costoso del proyecto ya que el tiempo de investigación fue el que más esfuerzo y dedicación requirió. Debido que la documentación de software está en proceso de desarrollo y las configuraciones son aun básicas, la búsqueda de paquetes de dependencia que no causen conflictos entre ellos fue una actividad importante, sin embargo las pruebas para conseguir un sistema estable fue lo que más tiempo consumió. Otro parámetro a analizar es el personal que lo desarrolló, ya que es mano de obra calificada y se han considerado 20 horas de trabajo por semana.

Se estimó un tiempo de 8 meses para conseguir la suficiente información para implementar y configurar el router, entonces calculando:

4 horas diarias de configuración y pruebas X 20 días hábiles al mes = 80 horas mensuales x 8 meses = 640 horas efectivas de investigación.

640 horas de investigación multiplicadas por 20.00 USD [44]la hora nos dan 12800 dólares por los 8 meses.

4.2.4 COSTO DE DOCUMENTACIÓN

El costo de la documentación lo evaluaremos como el tiempo empleado en elaborar este documento. Se estima un tiempo de 8 horas totales en transcribir toda la información.

4.2.5 COSTO TOTAL DEL PROYECTO

El costo total del proyecto de titulación (implementación del prototipo router ipv4/ipv6), se detalla en la tabla 4.3, el item más costoso es el que se invirtió en la investigación hasta obtener tanto el hardware como software confiables.

COSTOS	TIEMPO (Nº horas)	COSTO *hora (USD)	TOTAL (USD)
Costo equipo			304
Costo implementación	33	20	660
Costo investigación	640	20	12800
Costo documentación	8	10	80
TOTAL			13844

Tabla 4-3 Detalle de costo total del proyecto de titulación

Este valor inicial invertido, se estima recuperable comercializando el prototipo creado, para ello se debe determinar el precio final del producto, para establecer este valor se disminuyen las horas que se emplean en la implementación y configuración ya que se tiene la experiencia necesaria y se han desarrollado las pruebas que garantizan su funcionamiento.

[44] Se define un valor de 20 USD la hora de programación debido a que se consultó en 2 empresas el valor por hora para soporte tecnológico.

4.2.6 PRECIO DEL PROTOTIPO

El precio final se calculó incluyendo el costo del equipo y la implementación del ruteador.

Debido a la experiencia adquirida en la creación del ruteador las horas empleados en la implementación disminuyen notoriamente, se incrementó un costo de documentación ya que es importante y mandatorio entregar un manual básico de operación del equipo.

Descripción	TIEMPO (horas)	COSTO *hora (USD)	COSTO TOTAL (USD)
Costo del equipo			304
Instalación de sistema operativo	1	15	15
Recompilación de kernel y parches	3	15	45
Instalación de software de ruteo (XORP)	1	15	15
Instalación de software de calidad de servicio (CBQ.INIT)	1	15	15
Instalación de software de Ipsec (FREESWAN)	1	15	15
Configuración de XORP	2	20	40
Configuración de CBQ.INIT	2	20	40
Configuración de FREESWAN	2	20	40
Documentación			50
TOTAL			579

Tabla 4-4 Detalle precio final del prototipo

El costo del hardware hace que el prototipo aumente su valor, se debe considerar que si la empresa que adquiera el prototipo dispone del hardware adecuado para la implementación del router, se tomarán en cuenta únicamente los costos de implementación y de documentación, de tal forma como se señala en la tabla 4.5 el costo total por la creación del router dual sería:

Descripción	TIEMPO (horas)	COSTO *hora (USD)	COSTO TOTAL (USD)
Instalación de sistema operativo	1	15	15
Recompilación de kernel y parches	3	15	45
Instalación de software de ruteo (XORP)	1	15	15
Instalación de software de calidad de servicio (CBQ.INIT)	1	15	15
Instalación de software de Ipsec (FREESWAN)	1	15	15
Configuración de XORP	2	20	40

Configuración de CBQ.INIT	**2**	**20**	**40**
Configuración de FREESWAN	**2**	**20**	**40**
Documentación			**50**
TOTAL			**275**

Tabla 4-5 Detalle precio instalación y configuración del prototipo

4.3 COMPARACIÓN TÉCNICA CON LOS RUTEADORES EXISTENTES

En la comparación técnica fueron considerados equipos cuyas marcas son las significativas en el mercado como son Cisco, 3Com y Nortel.

Se comparó el prototipo implementado con dispositivos con parámetros de funcionalidad similares con el fin de identificar las ventajas y desventajas del prototipo creado. Cabe mencionar que algunas funcionalidades no están integradas en el ruteador Linux, sin embargo estas pueden ser implementadas añadiendo tarjetas y realizando en otros casos las respectivas configuraciones, estos parámetros no constituyen un limitante, ya que con la debida investigación pueden ser incorporados al prototipo.

Se debe señalar que los equipos con los que se va a realizar la comparación fueron creados por casas fabricantes que llevan años en el desarrollo de estos productos, por tal motivo presentan un diseño mejorado; como no es el caso del prototipo ya que está implementado en un computador, pero que tiene las características de hardware necesarias para realizar y cumplir con la funcionalidad para la que fue creado.

Se menciona a continuación los equipos y los parámetros a comparar:

- Cisco 2801.
- 3Com® Router 3018.
- Secure Router 1004 NORTEL.

Parámetros:

Memoria:

- Memoria RAM.
- Memoria Flash.

CPU:

- Procesador.

Tecnología de conectividad.- Se refiere a cómo va estar instalado el router.

Velocidad de transferencia de datos.- Expresada en bps.

Conexión de redes:

- Protocolo de direccionamiento.
- Protocolo de interconexión de datos.
- Red / Protocolo de transporte.
- Seguridad, se toma en cuenta si trabaja con IPsec, L2TP, GRE.
- Protocolo de gestión remota.
- Capacidad, si soporta túneles VPN IPSec.
- Características, se toma en cuenta si el router proporciona los siguientes servicios:
    - Protección firewall.
    - Soporte VLAN.
    - QoS, y los protocolos que utilice para proporcionar calidad de servicio por ejemplo: CAR, LAR, FIFO, GTS, PQ, CQ, WFQ, RED, WRED, LLQ.

Expansión / Conectividad:

- Interfaces.

Diverso:

- Algoritmo de cifrado.

Software:

- Sistema operativo proporcionado.

Parámetros de entorno:

- Temperatura mínima de funcionamiento.
- Temperatura máxima de funcionamiento.
- Ámbito de humedad de funcionamiento.

- Comparación Técnica

Especificaciones técnicas	CISCO 2801	3Com® Router MSR 30/60	NORTEL	ROUTER LINUX
Descripción del producto	Cisco 2801 Integrated Services Router	3com MSR 30/60	Secure Router 1004	
Tipo de dispositivo	Router	Router	Router	Router
Memoria				
Memoria RAM	128 MB (instalados) / 384 MB (máx.)	64 MB	256MB	1 GB (instalados)
Memoria Flash	64 MB (instalados) / 128 MB (máx.)	8 MB	32MB	
CPU			300MHz	
Tecnología de conectividad	Cableado	Cableado	Cableado	Cableado
Velocidad de transferencia de datos	100 Mbps			100 Mbps
Conexión de redes				
Protocolo de direccionamiento	OSPF, RIP-1, RIP-2, BGP, EIGRP	OSPF, RIP v1/v2, BGP-4, Routing estático	RIPv1, RIPv2, OSPF, BGP4, static routing, ECMP	OSPF, RIP v1/v2, BGP-4, Routing estático
Protocolo de interconexión de datos	Ethernet, Fast Ethernet	Ethernet	Ethernet, Fast Ethernet	Ethernet, Fast Ethernet

Red / Protocolo de transporte	IPSec	IP	IPSec	IPsec
Seguridad	IPSec	VPN (L2TP, GRE, IPSec), Firewall, ACLs, NAT, RADIUS, PAP/CHAP	IPSec	IPSec
Protocolo de gestión remota	SNMP 3	SNMP, HTTP	Telnet SSHv2, FTP/TFTP, SNMPv1, SNMPv2.	SNMP
Capacidad	Túneles VPN IPSec			Túneles VPN IPSec
Características	Cisco IOS Advanced IP services			
	protección firewall	protección firewall	protección firewall	protección firewall(no implementado)
	soporte VLAN	Soporte VLAN	soporte VLAN	soporte VLAN (no implementado)
	QoS (CAR, WFQ, CBWFQ, WRED, LLQ)	QoS (CAR, LAR, FIFO, GTS, PQ, CQ, WFQ, RED, WRED, LLQ)	RED, WRED, DiffServ, bandwidth guarantee/shaping, flow monitoring, Eight-level Priority Class Based Queuing (per IP address/subnets, ports, DSCP and ToS bits)	QoS (CBQ Class Based Queueing), SHAPING
Expansión / Conectividad				

Interfaces	2 x red - Ethernet 10Base-T/100Base-TX - RJ-45	Un puerto 10/100BASE-T,	2 x red - Ethernet 10Base-T/100Base-TX - RJ-45	3 x red - Ethernet 10Base-T /100Base-TX
	1 x red - auxiliar	1 x serie AUX	1 x red - auxiliar	
	1 x gestión - consola	1 x gestión - consola	1 x gestión - consola	
	1 x USB			
Diverso				
Algoritmo de cifrado	DES, Triple DES, AES		DES, 3DES, AES, SHA1, MD5	DES, 3DES, AES, SHA1
Software				
OS proporcionado	Cisco IOS Advanced IP services			Linux Debian - XORP
Parámetros de entorno				
Temperatura mínima de funcionamiento	0 °C	0 °C	0 °C	0 °C
Temperatura máxima de funcionamiento	40 °C	40 °C	40 °C	55 °C
Ámbito de humedad de funcionamiento	10 - 85%	0 to 95%	0 to 95%	

Tabla 4-6 Comparación técnica con ruteadores de similares características

Como se observa en la tabla 4.6, la mayoría de protocolos necesarios para realizar el enrutamiento están implementados en el prototipo; excepto los protocolos propietarios como lo es el EIGRP que es propiedad de cisco.

Los parámetros de memoria en este caso, exceden a la capacidad del resto de ruteadores, su uso es para almacenar y recuperar la información, si bien es cierto en un ruteador esto no es un punto crítico, las características actuales del mercado proporcionan buenos dispositivos de memoria a bajos costos.

Cada ruteador y el prototipo manejan la seguridad para el envío de paquetes con el protocolo IPsec; se debe mencionar además que para gestión remota el prototipo lo realiza con SNMP, el resto de ruteadores utilizan este protocolo y otros como el HTTP y mediante telnet.

La capacidad de túneles VPN IPsec solo la manejan el ruteador cisco y el prototipo. Características como protección de firewall soporte de VLAN vienen integradas en el resto de ruteadores, si bien en el prototipo no están incorporadas esto no implica que con las respectivas configuraciones no puedan ser añadidos, por lo que no representa ningún restricción en el prototipo.

La calidad de servicio se proporciona a través de CBQ Class Based Queueing, y SHAPIN, el resto de ruteadores manejan este mismo concepto.

En cuanto a las interfaces de red el equipo tiene 3 x red - Ethernet 10Base-T /100Base-TX, el resto de ruteadores poseen 2 sin embargo tienen la posibilidad de adicionar tarjetas, lo que implica un aumento en su costo.

Los parámetros de entorno fueron comparados con la temperatura de operación que soporta el MotherBoard del prototipo.

4.4 COMPARACIÓN ECONÓMICA CON LOS RUTEADORES EXISTENTES

Esta comparación como se menciona anteriormente fue realizada con equipos de similares características a las del prototipo, en la tabla 4.7 se distingue la diferencia de precios entre cada equipo.

ITEM	CISCO	3COM	NORTEL	ROUTER LINUX
Precio TOTAL (USD)	2420	3090	2018	579
(*) Soporte Técnico Anual (USD)	671	695	770	850

Tabla 4-7 Comparación económica con ruteadores de similares características

El precio del prototipo al igual que el resto de ruteadores no incluye la capacitación para su administración, este costo es adicional. En esta comparación se refleja que el valor del prototipo está por debajo de la mitad del costo de los otros ruteadores, sin embargo como se mencionó en la comparación técnica estos equipos poseen características adicionales no incorporadas en el prototipo.

El router dual fue creado como una solución de bajo costo que cumpla la función específica de enrutamiento, con un rendimiento adecuado por tanto su comercialización está destinada a pequeñas y medianas empresas.

Un aspecto importante es que el soporte técnico es más costoso debido a que existe menor cantidad de mano de obra calificada, aspecto que se puede mejorar

(*) *Para conocer el detalle de los valores se puede remitir al ANEXO H.*

con el tiempo ya que cada vez existe más personal calificado en herramientas Open Source.

CAPÍTULO 5.

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

1. Los altos costos del hardware y software de los fabricantes de dispositivos de enrutadores, ha ocasionado una búsqueda de opciones más económicas para reemplazar a dichos elementos principalmente por bajar los costos en especial en PYMES que no poseen los recursos para la adquisición de equipos tan costosos y que incluyen características y protocolos que en su mayoría no necesitan o no usan. Como respuesta a esta necesidad se desarrolló el presente proyecto de Titulación (Enrutador Linux) como una opción de enrutamiento basada en Software Libre (sin costo de licencias) que solventa las necesidades de empresas en crecimiento que no cuentan con el capital suficiente pero si con el personal capacitado para el manejo de sistema operativo Linux, que les permita optimizar los recursos y configurar el equipo en base a las necesidades específicas de la empresa.

2. El estándar IEEE830 proporciona una Especificación de Requisitos Software (ERS) que permite realizar un análisis y producir un documento con el detalle de funciones y restricciones del equipo, en cualquier tipo de proyecto una de las tareas más importantes es definir claramente los alcances, éstos representan las necesidades del producto que se debe desarrollar, basándose en estos resultados se procedió a la implementación del prototipo. Gracias a este análisis se define claramente la especificación de requisitos que el equipo debe cumplir, de esta manera evitamos que los costos de desarrollo puedan incrementarse o realizar

cambios en el equipo por no definir claramente sus funciones y restricciones.

3. Linux es un núcleo de Sistema Operativo muy robusto y versátil, pero requiere de un auspiciante (distribución) que le permita contar con un ambiente para utilizar todas sus capacidades de manera óptima, por lo que cada distribución genera un ambiente con características específicas que le permiten tener mejor desempeño para ciertas actividades por lo que basados en el análisis de IEEE830 se obtuvo que Debian es la distribución que permite obtener un mejor desempeño para el enrutador utilizando XORP, principalmente por su rapidez, desempeño, cantidad de paquetes y por la variada documentación que es posible encontrar en Internet o Bibliográficamente.

4. AHP (Proceso Analítico Jerárquico) es un método simple y flexible que construye un modelo jerárquico y permite de una manera eficiente organizar la información respecto a un problema, realiza comparaciones binarias y atribuye valores numéricos a los juicios (preferencias), usualmente hay dificultad cuando se va a tomar una decisión y se deben manejar muchos criterios o aspectos de un problema a la vez y se requiere la participación de varios actores como es el caso de priorizar los aspectos más importantes para la selección de hardware, gracias a este método se logró obtener un análisis de resultados en el cual se basó la selección y adquisición de hardware para la posterior implementación del ruteador Linux.

5. Luego de la debida selección de hardware, y el suficiente conocimiento del software a ser instalado, la reconfiguración del kernel es uno de los aspectos importantes en este proyecto, su correcta compilación y selección de módulos permitió obtener un sistema rápido y estable, en este caso se buscó activar los módulos de TCP/IP, Calidad de Servicio e IPsec, y

eliminar módulos como los de desarrollo que no serán utilizados. Como resultado de esta personalización se logró sacar un mayor provecho de las diferentes características que ofrece el software.

6. El software empleado tiene licencia GNU/GPL lo que implica un costo cero en su adquisición, con esto se logró disminuir en un 50% el precio final del enrutador permitiendo alcanzar uno de los objetivos planteados, ofrecer una solución menos costosa para pequeñas y medianas empresas, una de las fortalezas del software libre es la gran cantidad de información y soporte de comunidades que brindan ayuda en la solución de problemas, cabe mencionar que no todas estas soluciones son confiables ni inmediatas por lo que depende del grado de conocimiento del desarrollador, otra de las ventajas del software libre es el acceso al código fuente para la creación de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero.

7. Las especificaciones técnicas de ruteadores de marcas reconocidas en el mercado fueron comparadas con las del prototipo, con el fin de identificar sus fortalezas y debilidades, es necesario mencionar que el equipo creado pretende ser una solución a considerar en pequeñas y medianas empresas, algunas funcionalidades de los ruteadores de casas fabricantes no se presentan en el prototipo sin embargo no se consideran un limitante ya que podrán ser añadidas con el respectivo estudio, la incorporación de hardware y las debidas configuraciones.

8. Con respecto al software se puede destacar que el ahorro económico es muy elevado ya que ninguno de los paquetes utilizados tiene costo y lo que es más importante; que por la versatilidad del Sistema Operativo GNU Linux es posible añadir protocolos o funciones de monitoreo o seguridad adicionales sin requerir una actualización de sistema operativo o la instalación de módulos que tienen costos adicionales como es común en

los equipos de marca, aunque los costos de mano de obra podrían ser más altos.

9. El soporte para IPv6 e IPsec son una característica que convierte al enrutador Linux en un equipo con características que aseguran su funcionamiento para los futuros desarrollos y crecimientos que la globalización tecnológica obliga a soportar.

10. La posibilidad de complementar el enrutamiento tanto para paquetes IPv4 e IPv6 hacen del prototipo una buena solución tecnológica a ser aplicada hoy en día, debido a que en un futuro se estima la migración a esta nueva versión del protocolo de internet, con el fin de suplir el limitante del número de direcciones proporcionadas por el protocolo de Internet versión 4 y para aumentar las nuevas funciones incorporadas en el estándar IPv6, esta migración ofrece prescindir el uso de NAT (Network Address Translator) o traductores de direcciones de red, que conllevan a empresas con un número limitado de direcciones IP públicas la necesidad de usar direcciones privadas.

11. XORP es una plataforma de código abierto desarrollada gracias a la contribución económica de empresas a nivel mundial, la versión utilizada en el presente proyecto es la 1.6, que actualmente presenta un ambiente amigable que permite implementar los principales protocolos para enrutamiento. Gracias a XORP se obtuvo equipo completamente flexible brindando protocolos de enrutamiento IPv4 e IPv6 con una plataforma unificada para configurarlos.

12. El ruteador cuenta con módulos de código abierto para proporcionar seguridad en el envío de paquetes y la posibilidad de administrar el ancho de banda por medio del control de tráfico asignando prioridades. Con estas

configuraciones se proporciona a las Pequeñas y Medianas Empresas un sistema para enrutamiento más completo, robusto y funcional.

13. Linux a pesar de ser un sistema operativo muy versátil por sus potencialidades, no cuenta con interfaces de usuario que hagan de sus capacidades fáciles de utilizar, por lo que adicional al sistema operativo las comunidades de Software Libre han desarrollado interfaces que facilitan el uso y permiten aprovechar estas características de mejor manera. XORP por ejemplo es un desarrollo libre que utiliza el núcleo Linux para implementar una variedad de protocolos de enrutamiento complejos y de una forma mucho más amigable, de la misma manera se ha desarrollado scripts para la implementación de Reglas de Calidad de Servicio (cbq.init) y Seguridad IP (Raccon).

14. En el actual proyecto de titulación, una parte importante fue la selección de hardware para la implementación del prototipo, se optó por un método que presenta un sustento matemático, permite desglosar y analizar un problema e incluye la participación de diferentes personas o grupos de interés para generar un consenso; se realizó el estudio a una muestra de 29 personas encargadas de la adquisición de infraestructura de redes en pequeñas y medianas empresas de la ciudad de Quito, se tomaron en cuenta sus necesidades e intereses y se ordenaron esos elementos en un modelo jerárquico para priorizar el aspecto más importante en la selección de hardware, con esto se obtuvo un equipo que cumple con las expectativas y requerimientos de los administradores de infraestructura de red de las PYMES.

15. Con respecto al hardware se puede decir que es una parte vital para el funcionamiento adecuado del equipo, pero las características y requisitos están dados principalmente por la criticidad de los ambientes, la estimación del tráfico, la cantidad de rutas y los protocolos que van a ser

implementados, por lo que es de mucha importancia realizar un adecuado y exhaustivo dimensionamiento de capacidades, de acuerdo a lo propuesto, el enrutador Linux se desarrollo sobre un Computador Personal que es un equipo que abastece con mucha facilidad los requerimientos para tráfico intermedio gracias al Sistema Operativo Linux ya que tiene interesantes opciones de redundancia, y alta disponibilidad que permiten realizar configuraciones más robustas y seguras, con la ventaja de que los costos son mínimos en comparación con el hardware de enrutamiento existentes en el mercado.

16. El desarrollo e investigación de enrutadores en Linux es una opción que internacionalmente está creciendo con productos oficiales y costos relativamente más bajos, pero no son aceptados con facilidad por no ofrecer soporte técnico que permita atender rápidamente los requerimientos de los usuarios, por lo que su aceptación más que un problema de costos se convierte en un problema de confianza por la falta de mano de obra calificada que pueda ser solicitada frente a problemas críticos.

17. El gobierno ha anunciado por medio de mandatos oficiales su interés de utilizar software libre dentro de instituciones públicas con el objetivo de disminuir los costos de la adquisición de licencias, lo que ha permitido la apertura al desarrollo de herramientas de código abierto y a su vez la proliferación de empresas que ofrecen servicios de desarrollo y soporte sobre estas herramientas lo que sugiere una oportunidad muy alta para la aceptación de enrutadores basados en software libre a medida que el mercado note el ahorro en los costos y la fortaleza de los productos.

18. La implementación de soluciones basadas en software libre están generando una variedad de opciones que permiten ahorrar en costos principalmente de licencias, pero requieren de personal especializado para

realizar la configuración y el mantenimiento (servicios) de las soluciones y por la poca cantidad de mano de obra calificada en software libre los costos son mas altos en comparación a las soluciones en hardware populares.

19. Los desarrollos de software con respecto a interfaces gráficos y amigables para el manejo y configuración de servicios y aplicativos en Linux es muy bajo, pero los pocos que existen especialmente en herramientas para Firewall son de mucho éxito, por lo que la aceptación de enrutador Linux dependerá del desarrollos de un interfaz gráficos amigale que facilite el uso del enrutador.

5.2 RECOMENDACIONES

1. El enrutador Linux es una opción muy valiosa para la implementación de enrutamiento en pequeñas y medianas empresas, por lo que se recomienda la programación de herramientas gráficas que permitan manipular y administrar el funcionamiento de enrutamiento libre de una manera más amigable.

2. Se debe aprovechar la iniciativa gubernamental de apoyo a las herramientas libres para la formación de un centro de soporte confiable que brinde apoyo externo hacia empresas públicas y privadas y fortalecer la confianza en las herramientas y aplicaciones basadas en software libre.

3. Se recomienda realizar un ambiente de laboratorio más complejo y real que permita evaluar con parámetros reales de carga para conocer los umbrales con valores cuantitativos exactos.

4. Se recomienda la realización de un análisis previo del hardware sobre el que se implementará el prototipo para no exceder los gastos y para el adecuado funcionamiento del enrutador.

5. Se recomienda la creación de máquinas virtuales para la realización de pruebas previas a la implementación del equipo, con el fin de familiarizarse con el sistema operativo Linux y para la creación de varios escenarios.

6. Se recomienda la incorporación de herramientas gráficas como una consola de administración que permita manejar amigablemente las características del enrutador con el fin de hacerlo más competitivo en el mercado.

REFERENCIAS BIBLIOGRÁFICAS

LIBROS

[1] Folleto de TCP/IP Ing. Pablo Hidalgo.

[2] Estudio y configuración de calidad de servicio para protocolos ipv4 e ipv6 en una red de fibra óptica wdm. Sebastián Andrés Álvarez Moraga, Agustín José González Valenzuela 24 de julio de 2005.

[3] Introducción a Control del tráfico y a Calidad de servicio en GNU/Linux, Jon Latorre Martínez, Metabolik Bio Hacklab, mailto:moebius@etxea.net, versión 0.1, XX de mes de 2004.

[4] Redes Privadas Virtuales SEGURIDAD EN REDES TELEMÁTICAS, Luis Felipe Guerrero Ramírez, David Azañedo González, Curso 2004/2005.

http://asignaturas.diatel.upm.es/seguridad/trabajos/trabajos/curso%2004%2005/trabajo%20RPV.pdf

NORMAS

[5] Request for Comments: 2460, Diciembre 1998, Especificación Protocolo Internet, Versión 6 (IPv6), Pag 3, 4.

MANUALES

[6] Xorp

http://www.xorp.org

PÁGINAS WEB

[7] Wikipedia, Sistema Autónomo.

http://es.wikipedia.org/wiki/Sistema_aut%C3%B3nomo.

[8] Monografías, Direccionamiento IP.

http://www.monografias.com/trabajos29/direccionamiento-ip/direccionamiento-ip.shtml.

[9] Linux, Control de Tráfico.
05-Contro-Trafico-Linux-Fernando-David-Gomez.

[10] Monografias, Asociación de Seguridad.
http://www.alipso.com/monografias/protocoloipv6.

[11] http://es.wikipedia.org/wiki/Eclipse_(software)

[12] http://es.wikipedia.org/wiki/Colecci%C3%B3n_de_compiladores_GNU

[13] http://es.wikipedia.org/wiki/Planificaci%C3%B3n_Round-robin

BIBLIOGRAFÍA

LIBROS

[12] International Business Machines Corporation. (2005). Linux Network Administration I: TCP/IP and TCP/IP Services (course Code QLX07). Student Notebook. Linux Web.

[13] International Business Machines Corporation. (2005). Linux Network Administration II: Network Security and Firewalls (course Code QLX24). Student Notebook.Linux Web

MANUALES

[14] Cbq.init
http://beta.redes-linux.com/manuales/ancho_banda/control_ancho_banda_cbqinit.pdf.

[15] APH
http://en.wikipedia.org/wiki/Analytic_Hierarchy_Process

PÁGINAS WEB

[16] Debian
http://www.debian.org

[17] Cbq.init
http://listas.softwarelibre.cu/pipermail/linux-l/2005-March/047968.html

[18] Racoon
http://bookseguridad.blogspot.com/2008/05/ipsec-utilizando-racoon-en-linux.html

[19] Racoon
http://www.howtoforge.com/racoon_roadwarrior_vpn

[20] IPsec
http://www.ipsec-howto.org/t1.html

[21] Reconfiguración de Kernel
http://www.esdebian.org/articulos/24048/debian-kernel-26-como

[22] Reconfiguración de Kernel
http://www.mogaal.com/articulos/kernel-a-la-debian.html

[23] AHP
http://users.abo.fi/rfuller/sda18.pdf

[24] Cálculo de la muestra
http://www.hsa.es/id/investigacion-uai/uai_docs/muestreo.htm

[26] INEC
http://www.ecuadorencifras.com/cifras-inec/main.html

Distribuciones LINUX

[27] http://distrowatch.com/
[28] http://distrowatch.com/table.php?distribution=fedora
[29] http://fedoraproject.org/
[30] http://distrowatch.com/table.php?distribution=debian
[31] http://www.debian.org/
[32] http://distrowatch.com/table.php?distribution=pclinuxos
[33] http://www.pclinuxos.com/
[34] http://distrowatch.com/table.php?distribution=ubuntu
[35] http://www.ubuntu.com/
[36] http://distrowatch.com/table.php?distribution=gentoo

[37] http://www.gentoo.org/
[38] http://distrowatch.com/table.php?distribution=centos
[39] http://www.centos.org/
[40] http://distrowatch.com/table.php?distribution=mandiva
[41] http://www.mandriva.com/
[42] http://distrowatch.com/table.php?distribution=opensuse
[43] http://www.opensuse.org/
[44] http://distrowatch.com/table.php?distribution=slackware
[45] http://www.slackware.com/
[46] http://polishlinux.org/choose/

Software de enrutamiento

[47] www.freesco.org/
[48] www.xorp.org/
[49] www.coyotelinux.com/
[50] www.quagga.net
[51] www.zebra.org/
[52] www.openbsd.org/cgi-bin/man.cgi?query=routed&sektion=8
[53] www.linux-foundation.org/en/Net:Iproute2
[54] www.leaf.sourceforge.net/
[55] www.gated.org/

IPSec

[56] www.openswan.org/
[57] www.strongswan.org/
[58] www.freeswan.org/

Calidad de Servicio

[59] www.visolve.com/squid/whitepapers/qos.php
[60] www.usenet-forums.com/linux-networking/62639-bandwidth-shaping-cbq.html
[61] www.lists.netfilter.org/pipermail/netfilter/2001-November/028381.html
[62] www.rns-nis.co.yu/~mps/linux-tc.html