



# **UNIVERSIDAD DE QUINTANA ROO**

---

**División de Ciencias e Ingeniería**

Software para conmutación

Trabajo de Tesis para obtener el grado de

**Ingeniero en Redes**

**PRESENTA**

**Kinxoc Cano Lemus**

**Director de Tesis**

**Ing. Rubén Enrique González Elixavide**

**Asesores**

**Dr. Jaime Silverio Ortegón Aguilar**

**Lic. Luis Fernando Mis Ramírez**

**Chetumal, Quintana Roo, México, Julio de 2010.**



# **UNIVERSIDAD DE QUINTANA ROO**

---

## **División de Ciencias e Ingeniería**

Tesis elaborado bajo supervisión del Comité de Asesoría y aprobada como requisito parcial para obtener el grado de:

## **INGENIERO EN REDES**

Comité de Trabajo de Tesis

**Director:**

---

**Ing. Rubén Enrique González Elixavide**

**Asesor:**

---

**Dr. Jaime Silverio Ortegón Aguilar**

**Asesor:**

---

**Lic. Luis Fernando Mis Ramírez**

**Chetumal, Quintana Roo, México, Julio de 2010.**

## **Resumen**

En este proyecto de implementación de un software para el enrutamiento, se pretende utilizar un sistema informático que permita realizar una redirección entre las redes, dichas redes se crearán en un laboratorio, instalado en un sistema operativo de GNU/Linux, que puede ser Ubuntu o Debian. El objetivo principal constará de la instalación del programa de enrutamiento con la finalidad de realizar un comparativo en cuanto al funcionamiento y operación con un ruteador.

Al término de las pruebas que se llevarán a cabo en el desarrollo del proyecto, se tendrán los elementos necesarios para saber si el sistema de conmutación puede sustituir al ruteador, por la eficiencia del programa, o si solo serviría como un sistema de respaldo del mismo, en caso de daño.

## **Agradecimientos**

Cuando realicé estátesis, una de las partes más difíciles de escribir fueron los agradecimientos, porque no sabía con exactitud qué palabras usar para agradecerles a todas aquellas personas que me ayudaron a lograr este objetivo.

La creación de una tesis representa una etapa muy importante y enriquecedora en el camino que he recorrido. Toda la experiencia universitaria que he adquirido y en la elaboración del trabajo de la tesis, ha habido ciertas personas que merecen las gracias porque sin su valiosa aportación no hubiera sido posible este trabajo, y también muchas otras por haber plasmado su huella en mi camino.

A mis padres Ana María y Ricardo Ramón, por ser los mejores y estar conmigo incondicionalmente; por el apoyo y la formación que me brindaron; por fomentar en mí el deseo de saber, de conocer lo novedoso y abrirme las puertas al mundo ante mi curiosidad insaciable; Gracias porque sin ustedes y sus enseñanzas no estaría aquí, ni sería quien soy ahora, a ustedes les dedico este trabajo de tesis.

A mi abuelita, sé que me ves y estas orgullosa de mí; y a mi abuelito por su tolerancia y afecto.

A mi novia, que ha estado incondicionalmente a mi lado todo el tiempo.

A mis amigos y amigas, que siempre estuvieron, están y seguirán estando a mí lado brindándome su amistad, su cariño y apoyo.

A mis compañeros de estudio de la Universidad, con quienes construí conocimiento; compartiendo mañanas, tardes, noches de estudio y momentos de nerviosismo en exámenes parciales así como finales. Gracias por permitirme conocerlos y ser parte de su vida; por ayudarme al estar conmigo a lo largo de la carrera.

A mis compañeros de trabajo del Centro de Cómputo de la UQROO, que, me han ayudado a salir adelante en la vida diaria y adquirir el conocimiento necesario en el ámbito laboral.

A mis maestros, que estuvieron orientándome a lo largo de mis estudios de la carrera del profesional y de la ingeniería, por haberme soportado todos estos años. En especial a mis maestros: Ing. Rubén González, MC. Javier Vázquez, Dr. Jaime Ortegón.

A todos, *Gracias* por ayudarme a lograrlo.

## **Dedicatoria**

Les dedico esta tesis a mis padres que siempre han estado a mi lado, sin importar si iba bien o mal en la carrera, brindándome su apoyo en lo que me hiciera falta. Desde que tengo memoria ellos siempre han estado apoyándome, si me caía me ayudaban a levantarme, si lloraba me daban un hombro en el cual poder llorar, y por muchas cosas más en las cuales han estado a mi lado.

## ÍNDICE

Resumen.....	iii
Agradecimientos .....	iv
Dedicatoria .....	vi

## Capítulo 1

---

Introducción.....	1
Antecedentes .....	1
Definición del problema.....	2
Objetivos .....	3
Objetivo general .....	3
Objetivoparticulares.....	3
Justificación.....	4
Metodología .....	4
Alcances y restricciones.....	4

## Capítulo 2

---

Marco teórico .....	6
Marco contextual .....	8

## Capítulo 3

---

Desarrollo.....	11
Análisis .....	11
Diseño .....	12

Configuración del programa Quagga.....	13
Habilitando el reenvío en Debian.....	24
Habilitando el NAT en Debian .....	25
Verificando la configuración del servidor .....	27
Configuración de las máquinas para la red .....	27
Comparación .....	30

## Capítulo 4

---

Pruebas a realizar .....	36
Análisis de las pruebas realizadas .....	37
Análisis mediante D-ITG .....	40
Análisis con IPTraf.....	44
Análisis de transferencia .....	53

## Capítulo 5

---

Conclusiones.....	56
Bibliografía .....	60
Glosario.....	62
Abreviaturas.....	68
Anexo A .....	71
Instalación de Quagga.....	72
Anexo B .....	81
Daemons .....	82
Debian.conf .....	82
Ripd.conf .....	83
Zebra.conf .....	83

### Índice de Ilustraciones

Ilustración 1: La carpeta de Quagga .....	14
Ilustración 2: Lo que se modifica del archivo daemons.....	15
Ilustración 3: Modificando el archivo de daemons.....	15
Ilustración 4: Modificando el archivo debian.conf.....	15
Ilustración 5: Cómo está el archivo debían.conf.....	16
Ilustración 6: Cómo debe de quedar el archivo de debían.conf .....	16
Ilustración 7: Reiniciando el servicio de Quagga.....	17
Ilustración 8: Accediendo al Quagga para configurarlo.....	18
Ilustración 9: Cómo es la configuración del inicio del Quagga. ....	19
Ilustración 10: Cómo entramos a la configuración de las interfaces. ....	20
Ilustración 11: Si se quiere entrar por vty.....	23
Ilustración 12: Menú inicio del Windows. ....	28
Ilustración 13: Panel de control. ....	28
Ilustración 14: Propiedades de la interfaz de red. ....	29
Ilustración 15: Menú de configuración de la tarjeta de red.....	29
Ilustración 16: Configurar la IP a la máquina. ....	30
Ilustración 17: Programa alternativo para modalidad grafica del D-ITG. ....	33
Ilustración 18: La pestaña de Settings. ....	35
Ilustración 19: La pestaña de analyzer.....	35
Ilustración 20: Prueba 1 IPTraf, Ensamblada.....	45
Ilustración 21: Prueba 1 IPTraf, EVO. ....	45
Ilustración 22: Prueba 2 IPTraf, Ensamblada.....	46
Ilustración 23: Prueba 2 IPTraf, EVO. ....	46
Ilustración 24: Prueba 3 IPTraf, Ensamblada.....	47
Ilustración 25: Prueba 3 IPTraf, EVO. ....	47
Ilustración 26: Prueba 4 IPTraf, Ensamblada.....	48
Ilustración 27: Prueba 3 IPTraf, EVO. ....	48
Ilustración 28: Prueba 5 IPTraf, Ensamblada.....	49
Ilustración 29: Prueba 5 IPTraf, EVO. ....	49
Ilustración 30: Prueba 6 IPTraf, Ensamblada.....	50

Ilustración 31: Prueba 6 IPTraf, EVO.....	50
Ilustración 32: Prueba 7 IPTraf, Ensamblada.....	51
Ilustración 33: Prueba 7 IPTraf, EVO.....	51
Ilustración 34: Prueba 8 IPTraf, Ensamblada.....	52
Ilustración 35: Prueba 8 IPTraf, EVO.....	52
Ilustración 36: Pruebas con el ruteador 2811.....	54
Ilustración 37: Pruebas con la máquina EVO.....	55
Ilustración 38: Pruebas con la máquina ensamblada.....	55

## Índice de Tablas

Tabla 1: Características Principales de las Herramientas Evaluadas. ....	32
Tabla 2: Kbit/s. ....	40
Tabla 3: pkt/s.....	41
Tabla 4: Bytes recibidos.....	42
Tabla 5: Total de pkt. ....	43
Tabla 6: Transferencia ISO. ....	53
Tabla 7: Transferencia ISO y Archivos.....	54

## Índice de Figuras

Figura 1: Diagrama de la red.....	13
Figura 2: Kbit/s. ....	41
Figura 3: pkt/s. ....	42
Figura 4: Bytes recibidos.....	43
Figura 5: Total de pkt. ....	44

# Capítulo 1

---

## **Introducción**

Con este proyecto de Enrutamiento por Software, se pretende hacer una comparativa entre un ruteador Cisco y una máquina de escritorio con un programa de conmutación instalado, esta comparativa nos podrá decir si una máquina puede ser utilizada en lugar de un ruteador en una pequeña o mediana empresa, o como un sistema de respaldo en caso de daño del mismo.

De ser así, se podrá ahorrar dinero en equipos caros como son los ruteadores y en caso de una falla se podrá reducir el tiempo de inactividad del servicio, sustituyendo el equipo por uno de escritorio.

Para medir el desempeño del tráfico de paquetes entre los dispositivos, se utilizará un generador de tráfico, que permita obtener estadísticas de rendimiento.

## **Antecedentes**

En la Universidad de Quintana Roo (UQROO) existen 3 enlaces de red, los cuales se utilizan para distintas actividades.

Estos enlaces se dividen de la siguiente forma, el primero es un E3, el cual brinda la conexión a Internet, el segundo es un Multi-Protocolo mediante Etiquetas (*MPLS* por sus siglas en inglés) el cual conecta con la unidad de la UQROO de Cozumel con el Departamento de cómputo en Chetumal, y el tercer enlace sirve para la conexión de Internet 2 mediante una Red Privada Virtual (*VPN* por sus siglas en inglés).

Los 3 enlaces requieren de equipos de telecomunicaciones para poder conectarse a la red principal de Teléfonos de México (TELMEX), estos equipos se llaman ruteadores y según el enlace es el dispositivo que se requiere; existen

varias marcas y modelos. La UQROO tiene varios de la marca Cisco, uno para el enlace E3 modelo 3825 y dos más del modelo 2811 para los enlaces *MPLS* e Internet 2, por su fiabilidad y robustez, dado que en muy pocas ocasiones se dañan, soportan gran cantidad de tráfico además de contar con módulos de seguridad, de Calidad y Servicio. También se cuenta con un router Cisco 3640 como equipo de respaldo por si alguno de los anteriormente mencionados se llegara a dañar.

Un inconveniente es su alto costo, esto debido a que son equipos especializados y son cotizados generalmente en dólares; un ejemplo son los modelos de los equipos más pequeños que están alrededor de \$900 dólares, para pequeñas empresas, y por ende los precios suben hasta más de \$10,000 dólares para las empresas grandes. Los routers utilizados en la UQROO están alrededor de \$5,000 dólares, haciendo la conversión serían arriba de \$60,000 pesos aproximadamente según este el tipo de cambio; además del costo tenemos el inconveniente del tiempo de entrega en caso que se adquiera, el proveedor más cercano está en Cancún o Mérida, lo que hace que se pierda tiempo en la compra del equipo por el envío del mismo.

Tomando en cuenta los costos y la demora del envío, se pretende buscar una alternativa con este proyecto, con la finalidad de contar una opción en caso de alguna contingencia por daño del equipo o para la integración de algún centro alternativo a la red informática universitaria.

### **Definición del problema**

La Universidad de Quintana Roo cuenta con un router como respaldo, por el costo elevado de los mismos, si en algún momento se llegara a dañar alguno de estos 3 equipos, la Universidad podría cambiara uno de dos posibles, los cuales son el *MPLS* y el Internet 2 quedando fuera el enlace E3, porque el equipo no cuenta con el requerimiento mínimo para su funcionamiento, quedándose sin servicio de Internet por un lapso de tiempo indefinido, de aquí a que es

reemplazado, ya que en la ciudad no se cuenta con el soporte técnico para el mantenimiento correctivo de los equipos.

Otro punto a considerar es que es una universidad joven y dinámica con un crecimiento continuo y que demanda una mayor infraestructura tecnológica que sea confiable, que otorgue mejores y mayores servicios a la red de la institución. Se han creado nuevas unidades o centros que se encuentren fuera de las instalaciones principales de la UQROO en Chetumal, lo cual conllevaría a la adquisición de nuevos equipos que presten los servicios que la institución demanda, por lo que se hace necesario evaluar otras opciones que permita mantener los servicios activos y a un menor costo.

## **Objetivos**

### **Objetivo general**

Reducir el costo y tiempo en la adquisición de equipos de telecomunicaciones a través de aplicaciones libres.

### **Objetivos particulares**

1. Investigar la infraestructura lógica y física de los dispositivos de la red que conforman las Tecnologías de la Información y las Comunicación (TICs) de la Universidad de Quintana Roo.
2. Implementar los programas de conmutación.
3. Evaluar el rendimiento del software de conmutación en plataformas GNU/Linux y del router en un ambiente controlado.
4. Documentar los resultados obtenidos.
5. Recomendar lo necesario para implementar el nuevo software de conmutación en la Universidad.

## **Justificación**

Tomando en cuenta el crecimiento continuo de la Universidad de Quintana Roo, ya sea otra unidad o creación de centros fuera de la Universidad, se requeriría la adquisición de equipos de telecomunicaciones “ruteadores” de alto costo monetario.

Con el proyecto se pretende reducir costos de adquisición de dichos aparatos, en lugar de comprar un equipo de alto costo, se compraría una máquina de escritorio de menor costo, con la cual se realizaría el mismo trabajo.

O en dado caso podría servir como un sistema de respaldo, en caso de daño de algunos de estos equipos, ya que por lo mismo que son costosos, el tiempo de adquisición es largo, mínimo de una semana dado que el proveedor más cercano está en la ciudad de Mérida o Cancún y que no siempre lo tienen en sus bodegas.

## **Metodología**

### **Alcances y restricciones**

El alcance de este proyecto es evaluar las soluciones de enrutamiento, buscando alternativas más económicas, pero que sean confiables como una solución de respaldo, o incluso sustituir los equipos de enrutamiento en la nueva unidad Playa del Carmen y la División de Ciencias de la Salud.

Se implementaría una máquina de escritorio con un sistema GNU/Linux y un programa de enrutamiento, dando paso al cambio de los ruteadores por la máquina del proyecto. Ya sea como cambio permanente o como sistema de respaldo de los mismos.

Las posibles restricciones del proyecto pueden ser varias, la más crítica sería que se apruebe el presupuesto para la adquisición de los equipos de telecomunicaciones, con esto se comprarían los ruteadores dejando fuera el proyecto.

Otra posible restricción sería que el equipo del proyecto, con el que se cuenta para la implementación del proyecto, no cumpla con el desempeño mínimo que se requiere para la institución, dejando fuera la posibilidad de implementar dicho equipo en la red universitaria.

Por último el recurso humano asignado al proyecto, lo abandone por diversos motivos, ya sea personales, laborales, entre otras.

# Capítulo 2

---

## **Marco teórico**

Un ruteador es un dispositivo de conmutación de la capa 3 del modelo OSI, que permite la conexión de distintos segmentos de red y se usa muchísimo en las tecnologías Redes de Área Amplia (*WAN* por sus siglas en inglés), ya que permite coberturas amplias, acompañado con la tecnología que el proveedor del servicio que disponga puede ser *ATM*, *ADSL*, *HDSL,x-25*, *RDSI* entre otros. Igual se encarga de dirigir el tráfico hacia su destino en base a las tablas de ruteo y los protocolos que manejan. A través de esos protocolos, el ruteador conoce su entorno y se da cuenta de que redes son alcanzables por él, y por los equipos directamente conectados a él, y en cuantos saltos son alcanzables otro dispositivo.

Además, se encarga de recibir el tráfico destinado a una red, y de procesarlo para dejarlo pasar o no (funciones de *Firewall*), además, generalmente cuentan con listas de acceso (*ACLs*) que filtran los paquetes que pasan a través de ellos como medida de seguridad. Otra ventaja que brinda un ruteador es la Traducción de Dirección de Red (*NAT* por sus siglas en inglés), con la cual se puede dividir una red pública de una privada.

Dichos dispositivos no sólo sirven de salida a Internet, de hecho, Internet está conformado por muchos de ellos interconectados entre sí, sin computadoras en medio, y su aplicación es mucho más extensa que sólo manejar la línea de suscriptores digitales (*DSL* por sus siglas en inglés), un ruteador encuentra la mejor ruta, una puerta de salida, establece la comunicación hacia la red de tu proveedor de servicio de Internet, en donde se dirigirá el tráfico a su destino.

Los ruteadores se pueden encontrar desde unos cuantos cientos de dólares hasta miles de dólares, dependiendo de la marca y modelo al igual que de las tarjetas de expansión con las que se compran con el equipo.

Los problemas que se pueden encontrar es el mal dimensionamiento, hay ruteadores de muchos precios y funciones, aunque por lo general se piensa en equipo Cisco por ser de los mejores, la verdad es que si una empresa no necesita una gran capacidad de proceso, o de ancho de banda, un dispositivo pequeño puede ser la solución para cubrir sus necesidades, al igual se puede usar computadoras con GNU/Linux o Unix.

Como se mencionaba también se puede usar computadoras con GNU/Linux o Unix para emular el proceso de los ruteadores, esto con el fin de hacer más barato la creación de redes, al igual que su mantenimiento, dado que se puede usar desde una computadora convencional que funja a un servidor o a un enruteador con un OS libre o gratuito, esto se lograría con una computadora con buenos rendimientos y más con los nuevos procesadores, al igual que la memoria *RAM* y Disco Duro, de alta velocidad; en esos casos sólo se agregarían más tarjetas de Red para poder filtrar al tráfico de una red.

Con eso se reducirían los costos, en lugar de comprar equipos de ruteo o servidores carísimos, que una empresa o institución no requiere en ocasiones, de tal forma se puede ahorrar hasta un 40% aproximadamente del precio final, esto debido a que ya no se adquirirían 2 ruteadores y en lugar de estos se adquirirían dos máquinas equipadas con mejor memoria, procesador y disco duro.

Las herramientas de conmutación, son programas que trabajan bajo las plataformas de *GNU/Linux* o *Unix* para brindarle más seguridad a la misma herramienta o software, la mayoría de las veces son gratuitas y cuentan con la documentación necesaria para su instalación y configuración.

Estas herramientas tienen la facilidad de sustituir el trabajo de los equipos de enrutamiento, desde poder crear un *Firewall* (corta fuegos) mediante las *ACLs*, así como filtrado de paquetes, aislamiento de computadoras y puertos que puedan dañar la integridad de la red. También interconectar varias redes, en

cuanto a su eficiencia dependerá de la computadora en donde esté funcionando, mientras mejor desempeño tenga la máquina más elevado será el costo.

El software libre puede manejar varios protocolos de enrutamiento como *RIP*, *IGRP*, *EIGRP*, *OSPF*, *BGP*, entre otros protocolos, sus conexiones son las mismas que las del ruteador, solo que se manejan unas tarjetas de red las cuales se adquieren para la PC para el puerto *PCI* de la computadora o algún otro disponible y se piden como tarjetas *Wan Interface Card(WIC)*.

Por lo general los mismos OS de GNU/Linux o Unix pueden fungir el funcionamiento de estos equipos de telecomunicaciones, pero en ocasiones eran algo limitados para lo que se requería, por consecuente salieron unos programas dedicados a la conmutación de paquetes, con las cuales se harán las pruebas para ver cuál es la mejor para la Universidad de Quintana Roo, dado que el tráfico en la red en dicha institución es bastante cargado, aun contando con 10 Mb de ancho de banda dedicado, porque se tienen conectados a la red aproximadamente unas 800 computadoras de escritorio y más de 100 *Laptops* en la red inalámbrica a todos estos equipos se les da acceso a Internet

### **Marco contextual**

El proyecto tendrá lugar en la Universidad de Quintana Roo, la cual tiene más de 3,700 alumnos inscritos en el ciclo otoño 2010, 168 docentes de tiempo completo, 249 por asignatura y 286 administrativos; a los 2 últimos se les brinda acceso a los sistemas institucionales, los cuales están en la intranet de la institución, de igual forma tienen Internet gracias a la red universitaria, en el caso de los alumnos también pueden entrar al Internet y los sistemas de la UQROO, esto lo pueden hacer desde la biblioteca de la universidad o si cuentan con una máquina con acceso al WIFI.

Esto se logra a través del enlace dedicado de Internet, con el cual se cuenta en la Universidad, el enlace es un E3 a 10 Mb, administrado por un

*Firewall(Checkpoint)*, este equipo permite administrar este ancho de banda mediante el sistema de calidad de servicio, dicho enlace llega a la Universidad por fibra óptica a la acometida de TELMEX, ahí pasa a sus equipos y consecuentemente a los ruteadores de la Universidad.

En el caso de la unidad de Chetumal, Cozumel y en unos meses la unidad de Playa del Carmen, se conectan internamente mediante un enlace MPLS, este enlace logra crear una red tipo MAN entre las unidades, con el fin de tener una interconexión entre ellas mismas, como si estuviesen en el mismo lugar, así pueden acceder a los servicios internos sin problemas. Dicho enlace, al igual que el E3, llega a un ruteador el cual pasa después al Firewall para su administración.

En el Firewall se hace una división virtual y dicha división consta de dos partes, la primera es la red de la intranet de la universidad y la segunda la red que llega a través del enlace E3, el cual es el Internet. Esta segmentación sirve para seguridad de la universidad de posibles ataques externos; al igual de esta forma se puede tener los servidores a los cuales se tiene acceso desde fuera de la universidad, y los de control interno que son los sistemas institucionales ya sea el sistema de finanzas, los datos internos de la universidad, entre otras cosas.

Otra bondad del Firewall para la unidad de Chetumal y Cozumel es que permite una segmentación por edificio, y cada unidad tiene un bloque de direcciones IP distintos, con el fin de que no choquen entre ellos en la red MAN, tomando en cuenta que es una gran red informática administrada desde el Checkpoint, cuando se integre la unidad de Playa del Carmen, tendrá su propio bloque de direcciones de red y será agregada a la RIU.

Otro enlace con el que cuenta la universidad es una VPN conectada a Cancún, para el acceso a Internet 2, mediante este enlace la Universidad de Quintana Roo se comunica con otras universidades del mundo, ya sea para una

videoconferencia, acceso a bases de datos de otras instituciones que cuenten con el Internet 2, entre otras cosas.

Estos tres enlaces llegan a 3 ruteadores distintos, uno por cada enlace en la unidad Chetumal, más otros 2 que están en la unidad Cozumel para los enlaces de Internet y MPLS, y 2 en la Unidad Playa del Carmen cuando se integre próximamente; es decir, 7 equipos trabajando los 365 días del año, con un alto costo cada uno de los equipos.

# Capítulo 3

---

## Desarrollo

### Análisis

Con el inicio del proyecto se tiene que averiguar si existen y cuáles serían los programas que puedan hacer la función de enrutamiento.

Por tal motivo se investigó en la Internet y se encontraron 5 programas que realizan esta función, los cuáles son: Quagga, XORP, Zebra, BIRD, OpenBGPD, cada uno con ventajas y desventajas sobre los otros.

La primera que quedó descartada fue Zebra, ya que actualmente no tiene soporte y su última actualización fue en septiembre de 2005.

El siguiente que se eliminó es OpenBGPD, dado que sólo trabaja con Unix y únicamente maneja el protocolo *GBP* versión D, su última actualización es de noviembre de 2009.

XORP tiene soporte para una amplia gama de protocolos y sistemas operativos, los inconvenientes son que no soporta el protocolo *RIP* y que su última versión es de enero de 2009, otro inconveniente es que el programa ya no tendrá soporte de la empresa XORP Inc., quedando como código libre.

En cuarto lugar quedó BIRD, siendo uno de los más recientemente actualizados en agosto de 2010, maneja varios protocolos, su único inconveniente es que solo es soportado para Unix.

Y por último tenemos Quagga, con varios protocolos, actualizado constantemente, tiene soporte para GNU/Linux y Unix, además de ser un ramificación de Zebra, como complemento puede trabajar con las versiones de IPv4 y la IPv6.

Teniendo en cuenta estos 5 programas se optó por el Quagga, por ser actualizado constantemente, variedad en protocolos, soporte para el programa, y poder funcionar en varios sistemas operativos.

La otra parte del análisis es elegir el mejor sistema operativo, de los que se soportan por el Quagga.

Los OS que soporta el Quagga son aquellos basados en Unix, particularmente FreeBSD, Solaris, NetBSD y las diferentes implementaciones de GNU/Linux como Debian, CentOS, RedHat, Ubuntu, entre otros. Los sistemas GNU/Linux antes mencionados son de los más usados en la actualidad, todos pueden trabajar con Quagga, pero solo Debian tiene soporte y paquete de instalación específicamente para el sistema operativo. Por tal motivo se optará por el Debian como sistema operativo del proyecto.

## **Diseño**

Para poder realizar el proyecto se utilizarán 4 cables *straight-through* de red categoría 5e con 2 conectores RJ-45 cada uno y un quinto cable *crossover*, 2 equipos Switch de la marca Cisco modelo 2950 para interconectar con las computadoras, de igual forma se utilizaron 2 ruteadores de la marca Cisco modelo 2811, 2 máquinas de escritorio, la primera es una máquina COMPAQ del modelo EVO, y la otra es una ensamblada con lo último en tecnología, los equipos antes mencionados tendrán el sistema Debian con el programa Quagga, y por último 2 máquinas más, una Laptop de la marca Acer modelo Aspire 5520 y la otra es de escritorio, HP modelo DC7600, las cuales servirán para el envío y recepción de paquetes, con el fin de cargar el ancho de banda de los equipos a probar, para ver que desempeño se logra de cada uno.

La organización de los equipos queda de la siguiente forma: primero está la Laptop Acer, con un cable de red al *Switch* 2950, el cual se conecta al ruteador 2811 mediante otro cable de red, dicho dispositivo se configuro previamente. El

ruteador 2811 se conectará mediante un cable de red cruzado a uno de los 3 equipos con la que se realizarán las pruebas, ya sean el segundo ruteador 2811, la máquina ensamblada o la COMPAQ, los cuales se conectan al segundo *Switch* 2950 con el tercer cable de red. Finalmente, se conectará el *Switch* a la máquina HP con el 4 cable de red. Cabe mencionar que los 4 cables de red que se conectan a los 2 *Switches* con *straight-through* porque los equipos ya pueden hacer el cambio de la configuración del cableado.

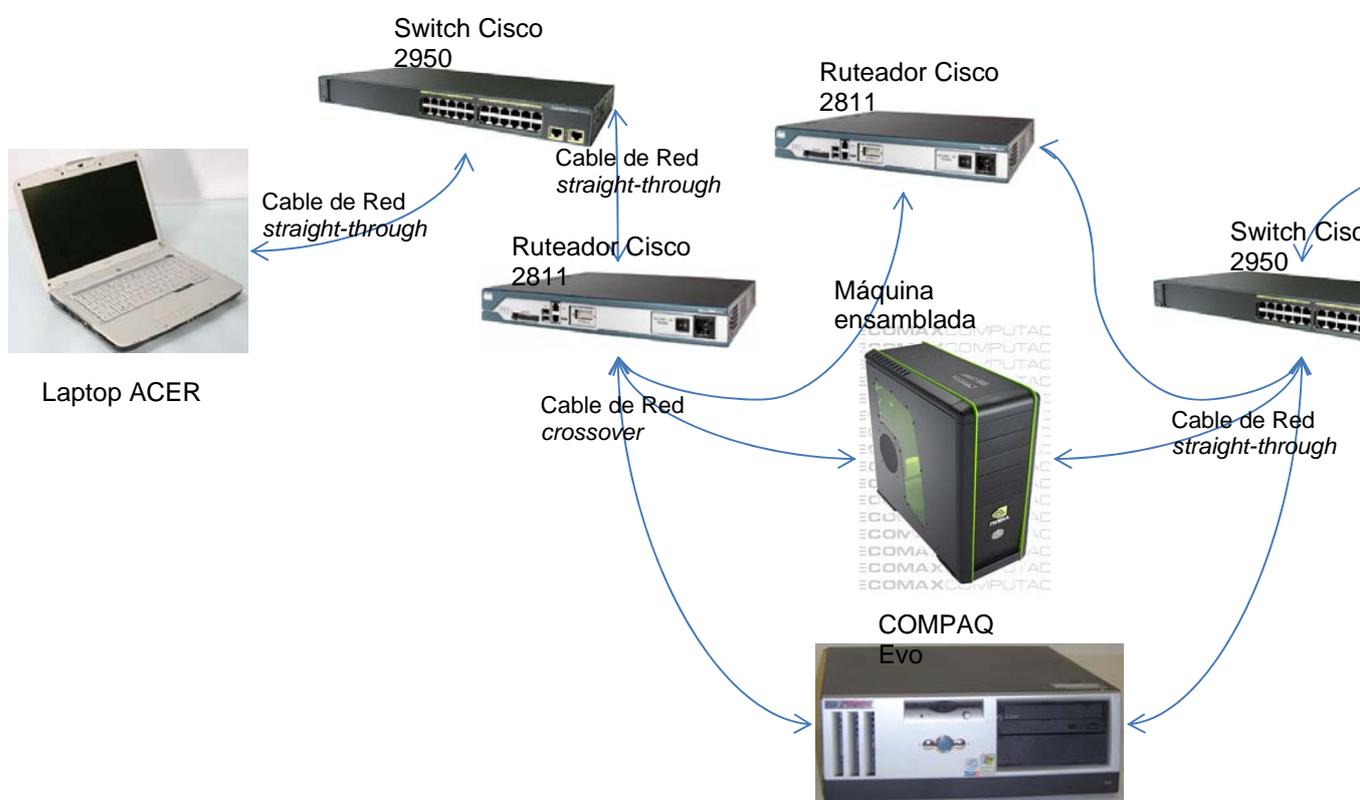


Figura 1: Diagrama de la red

### Configuración del programa Quagga

Para la instalación de Quagga, el lector se puede referir al anexo A. Previo a la configuración, se deben copiar los archivos de la carpeta `/usr/share/doc/quagga/examples/` a la carpeta `/etc/quagga/`.

Renombramos los ejemplos de configuración recién copiados, únicamente renombraremos los archivos que vallamos a utilizar en este caso serían **zebra.conf.sample** y **ripd.conf.sample** a los cuales les quitamos la sección **.sample**, esto porque los archivos antes mencionados ya tienen código para usar y activar Quagga con la configuración por defecto. Para los archivos completos, con la configuración final, el lector se puede referir al anexo B. Todos los comandos que se presentan a lo largo del documento se deben introducir como administrador (root o súper usuario). La configuración se puede hacer con los siguientes comandos<sup>1</sup>:

```
cd /etc/quagga/  
cpzebra.conf.sample zebra.conf  
cpripd.conf.sample ripd.conf
```

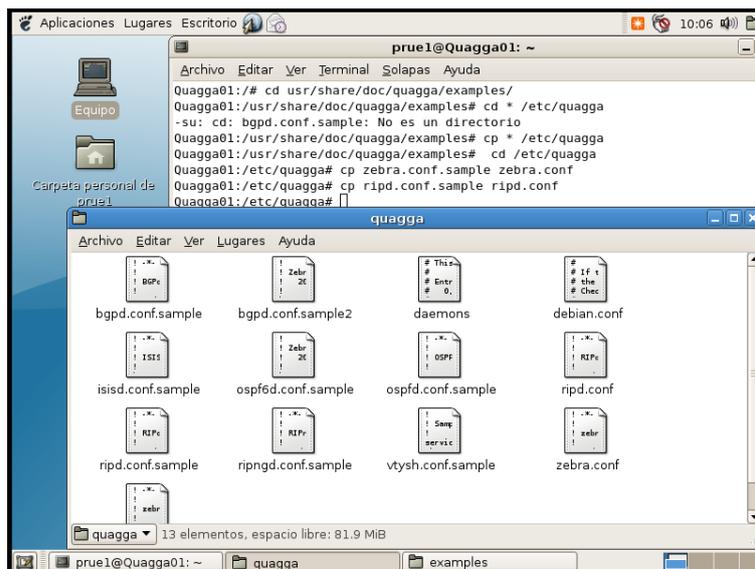


Ilustración 1: La carpeta de Quagga

Con esto indicamos que Quagga arranque los demonios **zebra** y **ripd**, cada vez que Quagga inicie con el servidor.

<sup>1</sup>Las secciones de texto que tengan el formato de letra `courier new` se refieren a código o comandos.

Configuramos los demonios para que active **Zebra**, y **rip**, aunque se hace de igual modo para los demás protocolos, ponemos:

```
vimdaemons
```

Solo editamos las líneas para activar el demonio de **zebra** y **rip**, en lugar de *no* ponemos *yes*<sup>2</sup>:

```
zebra=yes
```

```
ripd=yes
```

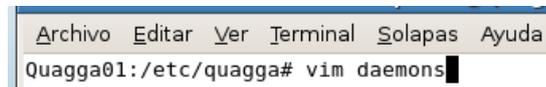


Ilustración 2: Lo que se modifica del archivo daemons.

#	zebra=no	zebra=yes
	bgpd=no	bgpd=no
	ospfd=no	ospfd=no
	ospf6d=no	ospf6d=no
	ripd=no	ripd=yes
	ripngd=no	ripngd=no
	isisd=no	isisd=no
		:wq!

Ilustración 3: Modificando el archivo de daemons.

Si se quieren usar otros demonios solo se modifica el protocolo de *no* al *yes*.

De igual forma modificamos el registro **debian.conf**, esto para agregar las direcciones IP de nuestra red, con los cuales va a trabajar el servidor.

```
vimdebian.conf
```

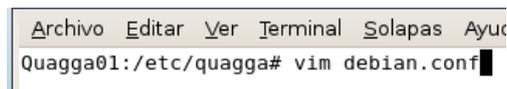


Ilustración 4: Modificando el archivo debian.conf.

---

<sup>2</sup>Para salir de vim apretamos las siguiente teclas: **ESC :wq!**, con esto indicamos **wwriteqquit** y ! forzar la opciones

```
Zebra_options=" -daemon -A 127.0.0.1"
```

```
~
vtysh_enable=yes
zebra_options=" --daemon -A 127.0.0.1"
bgpd_options=" --daemon -A 127.0.0.1"
ospfd_options=" --daemon -A 127.0.0.1"
ospf6d_options=" --daemon -A ::1"
ripd_options=" --daemon -A 127.0.0.1"
ripngd_options=" --daemon -A ::1"
isisd_options=" --daemon -A 127.0.0.1"
~
```

#### Ilustración 5: Cómo está el archivo debían.conf.

En este caso quedaría de la siguiente forma dado que contamos con 3 interfaces de red y cada una tiene dirección IP:

```
Zebra_options=" -daemon -A 127.0.0.1 192.168.8.157 10.10.100.254
10.10.200.254"
```

```
~
vtysh_enable=yes
zebra_options=" --daemon -A 127.0.0.1 192.168.8.157 10.10.100.254 10.10.200.254"
bgpd_options=" --daemon -A 127.0.0.1"
ospfd_options=" --daemon -A 127.0.0.1"
ospf6d_options=" --daemon -A ::1"
ripd_options=" --daemon -A 127.0.0.1"
ripngd_options=" --daemon -A ::1"
isisd_options=" --daemon -A 127.0.0.1"
~
```

#### Ilustración 6: Cómo debe de quedar el archivo de debían.conf

Para que cargue la modificación antes realizada reiniciamos Quagga, para lograr eso tenemos varias formas:

Con esta línea detenemos e iniciamos Quagga de una sola vez:

```
/etc/init.d/quaggarestart
```

Para solo iniciar el servicio de quagga

```
/etc/init.d/quagga start
```

Para detenerlo

```
/etc/init.d/quagga stop
```

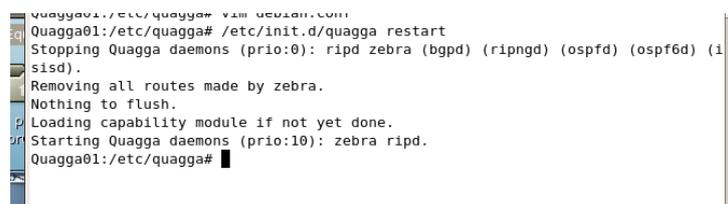
Ya que pusimos **/etc/init.d/quagga restart** nos mostrará las siguientes líneas:

```
#Stopping Quagga daemons (prio:0): (ripd) (zebra) (bgpd) (ripngd)
(ospfd) (ospf6d) (isisd).
```

```
Removing all routes made by zebra.
```

```
Nothing to flush.
```

```
Loading capability module if not yet done. Starting Quagga daemons
(prio:10): zebra ripd.#
```



```
Quagga01:/etc/quagga# /etc/init.d/quagga restart
Stopping Quagga daemons (prio:0): ripd zebra (bgpd) (ripngd) (ospfd) (ospf6d) (i
sisd).
Removing all routes made by zebra.
Nothing to flush.
Loading capability module if not yet done.
Starting Quagga daemons (prio:10): zebra ripd.
Quagga01:/etc/quagga#
```

**Ilustración 7: Reiniciando el servicio de Quagga.**

Ahora podremos acceder por separado con una interfaz lineal a cada uno de los demonios **zebra** y **ripd**.

Para acceder a Zebra

(*Password* por defecto zebra, para ambos demonios):

```
Telnet localhost 2601
```

(Para acceder al demonio de RIP *ripd*):

```
Telnet localhost 2602
```

Configurando Zebra

Zebra es el demonio gerente. Los demonios que se encargan de los protocolos de enrutamiento son: **ripd**, **ripngd**, **ospfd**, **ospf6d**, **bgpd**.

Podemos configurar Zebra editando el fichero de configuración **zebra.conf** o bien accediendo al demonio y por diversos comandos. Antes que nada con el signo de interrogación “?” obtenemos una ayuda contextual de las posibilidades que se va clasificando, es decir, si ponemos “?” nada más y le damos en la

tecla de enter, nos dice los posibles comandos que tenemos, pero si escribimos:

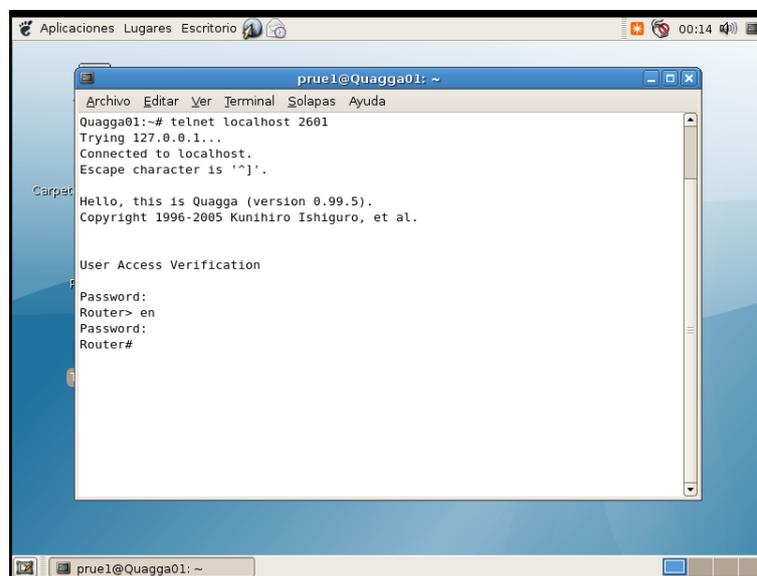
debug ?

Nos informa sobre los modificadores del comando **debug** en concreto. Esto es importante que quede claro, porque nos va a servir de mucho al principio, hasta que vayamos aprendiendo las principales instrucciones.

Un ejemplo, cambiemos el nombre de host del ruteador, ¿para qué? Imaginemos una red de más de 10,000 computadores, dividida en subredes comunicadas entre unos cuantos ruteadores, ¿no sería útil que cada ruteador tuviera un **prompt** distinto? Así podríamos configurar varios a la vez sin tener que recordar que **vty** corresponde a cada Router.

Iniciamos haciendo un telnet a la interface de la máquina al puerto 2601:

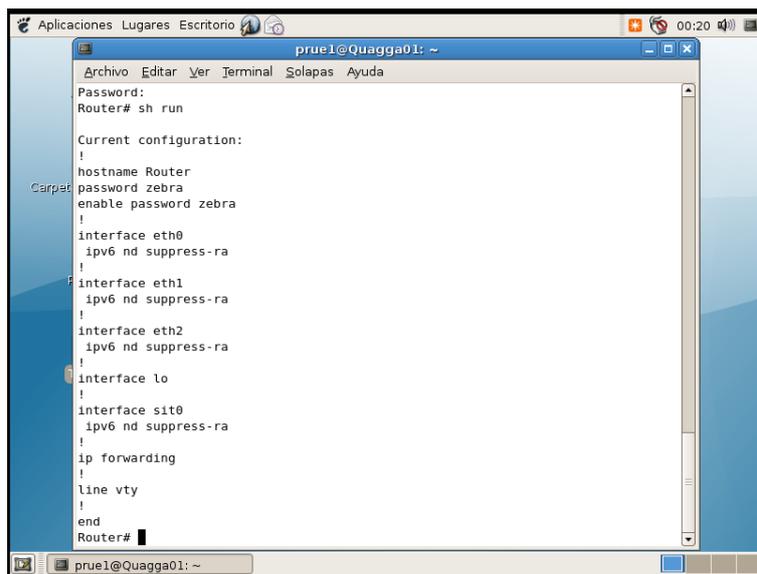
```
Telnet localhost 2601
```



**Ilustración 8: Accediendo al Quagga para configurarlo.**

Con esto entramos a la configuración de **ripd**, para agregar la configuración que requiera el ruteador. Una vez en el sistema, estamos en modo usuario, pero

solo se puede ver la configuración que tiene elruteador, mediante el comando **Show run**; para poder empezar con la configurar el ruteador, tenemos que acceder al modo súper usuario mediante el comando **enable**, a continuación pide un *Password* el cual es **zebra**.

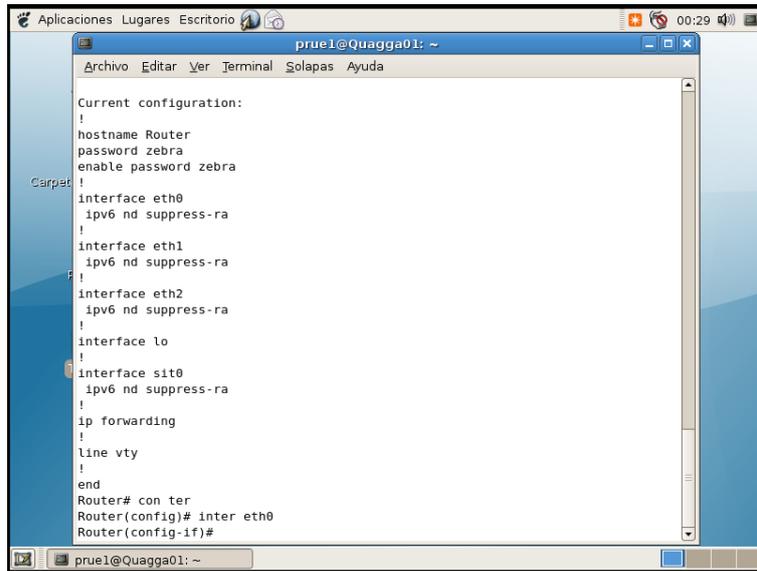


```
prue1@Quagga01: ~
Archivo Editar Ver Terminal Solapas Ayuda
Password:
Router# sh run

Current configuration:
!
hostname Router
password zebra
enable password zebra
!
interface eth0
ipv6 nd suppress-ra
!
interface eth1
ipv6 nd suppress-ra
!
interface eth2
ipv6 nd suppress-ra
!
interface lo
!
interface sit0
ipv6 nd suppress-ra
!
ip forwarding
!
line vty
!
end
Router#
```

**Ilustración 9: Cómo es la configuración del inicio del Quagga.**

Ya estando como súper usuario, se agregan más opciones de comandos para la configuración del ruteador; así como las anteriores existen 3 modos más, el siguiente es el modo de configuración global, en el cual se puede cambiar el *password* de acceso al **ripd** y al **enable**, también el nombre del ruteador para poder identificarlo, otro modo es el de interface, en el cual agregamos los tipo de redes con las que cuente, ya sea **ppp**, **adsl**, **eth** entre otras, se puede agregar el IP, la máscara de cada una de las redes que contenga su servidor, y por último el modo **vty**, en él se puede configurar el acceso remoto a su servidor para su manipulación de Quagga.



```
prue1@Quagga01: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda

Current configuration:
!
hostname Router
password zebra
enable password zebra
!
interface eth0
  ipv6 nd suppress-ra
!
interface eth1
  ipv6 nd suppress-ra
!
interface eth2
  ipv6 nd suppress-ra
!
interface lo
!
interface sit0
  ipv6 nd suppress-ra
!
ip forwarding
!
line vty
!
end
Router# con ter
Router(config)# inter eth0
Router(config-if)#
```

Ilustración 10: Cómo entramos a la configuración de las interfaces.

Algunos podrán pensar que se parece mucho a la interface que se utiliza en los equipos Cisco, pues sí; la sintaxis es en un 95% aproximadamente idéntica a la de un Cisco, aunque Quagga no tiene algunas configuraciones que un ruteador Cisco dispone.

A continuación se pondrán los comandos para realizar la configuración del demonio **zebra.conf** al igual que en un ruteador Cisco, se permiten las abreviaciones de los comandos. Vale la pena recordar que los comandos se deben introducir como súper usuario.

```
Debian:~#telnet localhost 2601
```

El puerto que se utilizará es el 2601.

```
Password:
```

El *password* por default es **zebra**.

```
Router>
```

Está en el modo usuario.

```
Router>en
```

Acceso a modo súper usuario.

Password:

Igual por default es **zebra**.

Router#

Cambio de > a #, haciendo la diferencia del Prompt, de esta manera se indica que se encuentra en modo súper usuario.

Router#conf ter

Modo de configuración global.

Router(config)#hostname Quagga

Asigna un nombre al equipo.

Quagga(config)#password quagga

Contraseña que se usará para ingresar al Quagga.

Quagga(config)#enable password quagga

Contraseña que se usará para ingresar en el modo EXEC privilegiado.

Quagga(config)#line vty

Para poder tener acceso mediante Telnet desde una máquina en la red.

Quagga(config-line)#login

Permite la verificación de la contraseña en la línea.

Quagga(config-line)#exit

Salimos del modo de configuración de Telnet.

Quagga(config)#inter eth0

Modo de configuración de interfaces.

Quagga(config-if)#ip address 192.168.8.157/24

Configura la IP y máscara.

Quagga(config-if)#description "salida a Internet"

Anotación propia del administrador para diferenciar las interfaces.

Quagga (config-if)#no shutdown

Activa la interface que se configuró.

Quagga(config)#inter eth1

Modo de configuración de interfaces.

Quagga(config-if)#ip address 10.10.100.254/24

Configura la IP y máscara.

```
Quagga(config-if)#description "Red Lan 1"
```

Anotación propia del administrador para diferenciar las interfaces.

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

Activa la interface que se configuró.

```
Quagga(config)#inter eth2
```

Modo de configuración de interfaces.

```
Quagga(config-if)#ip address 10.10.200.254/24
```

Configura la IP y máscara.

```
Quagga(config-if)#description "Red Lan 2"
```

Anotación propia del administrador para diferenciar las interfaces.

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

Activa la interface que se configuró.

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

Ponemos *exit* para salir de un modo a otro.

```
Quagga(config)#iproute 0.0.0.0 0.0.0.0 192.168.8.254
```

Se crea una ruta estática para el ruteador.

```
Quagga(config)#router-id 192.168.8.157
```

Identifica el ruteador con una dirección ip.

```
Quagga(config)#exit
```

```
Quagga#wr
```

Este comando guarda la configuración insertada anteriormente.

```
Configuration saved to /etc/quagga/zebra.conf
```

```
Quagga#exit
```

Salimos totalmente de la configuración y regresamos a la ventana del OS en la que se encontraba en un inicio.

En caso de utilizar el servicio de *vty*, se tiene que agregar las direcciones IP en el servicio *debían.conf* de las máquinas que se van a conectar al servidor.

```

vtysh_enable=yes
zebra_options=" --daemon -A 127.0.0.1 192.168.8.157 10.10.100.254 10.10.200.254"
bgpd_options=" --daemon -A 127.0.0.1"
ospfd_options=" --daemon -A 127.0.0.1"
ospf6d_options=" --daemon -A ::1"
ripd_options=" --daemon -A 127.0.0.1 192.168.8.12"
ripngd_options=" --daemon -A ::1"
isisd_options=" --daemon -A 127.0.0.1"
~

```

Ilustración 11: Si se quiere entrar por vty.

Con esto terminamos la configuración del **zebra.conf**. A continuación se proseguirá con el archivo **ripd.conf**, en donde se pondrá otra parte de la configuración del Quagga, en ella se configurará el protocolo de enrutamiento **RIP**.

Debian:~#

Estando en modo súper usuario.

Debian:~#telnet localhost 2602

El puerto que se utilizará es el 2602.

password:

El password por default es **zebra**.

Quagga>

Está en el modo usuario.

Quagga>en

Acceso a modo súper usuario no pide password.

Quagga#conf ter

Accedamos al modo de configuración global.

Quagga(config)#hostname Quagga\_rip

Quagga(config)#password Quagga

Quagga(config)#enable password quagga

Quagga(config)#router rip

Se pone router y se agrega el protocolo a utilizar en este caso **RIP**

Quagga(config-router)#network 192.168.8.0/24

Network especifica los segmentos e interfaces directamente conectadas al ruteador que serán anunciadas por **RIP**.

Quagga(config-router)#network eth0

Quagga(config-router)#network 10.10.100.0/24

Quagga(config-router)#network eth1

```
Quagga(config-router)#network 10.10.200.0/24
Quagga(config-router)#network eth2
Quagga(config-router)#version 2
```

Versión 1 ó 2, para especificar la versión de *RIP* que se va a utilizar, si no se especifica la versión, el software *IOS* adopta como opción predeterminada el envío de *RIP* versión 1.

```
Quagga(config-router)#default-information originate
```

Le manda a los ruteadores cuál es la puerta de salida por default del ruteador

```
Quagga(config-router)#Exit
Quagga(config)#Exit
Router#wr
```

Guarda la configuración que se agregó o modificó en el Quagga

```
Configuration saved to /etc/quagga/ripd.conf
```

Es el mensaje que muestra de que se guardó correctamente el archivo de igual manera muestra que archivo se guardó.

```
Quagga#Exit
```

Terminado estos pasos el servidor está casi listo.

## Habilitando el reenvío en Debian

Continuamos con un punto crítico para el servidor, el reenvío de paquetes de una interface a otra, por parte del OS Debian, tenemos 2 opciones, la primera que solo es temporal, sirve una vez desde que se pone hasta que se reinicie o apague el servidor, la otra es permanente dado que se guarda en un archivo del OS, a continuación se muestran las dos opciones.

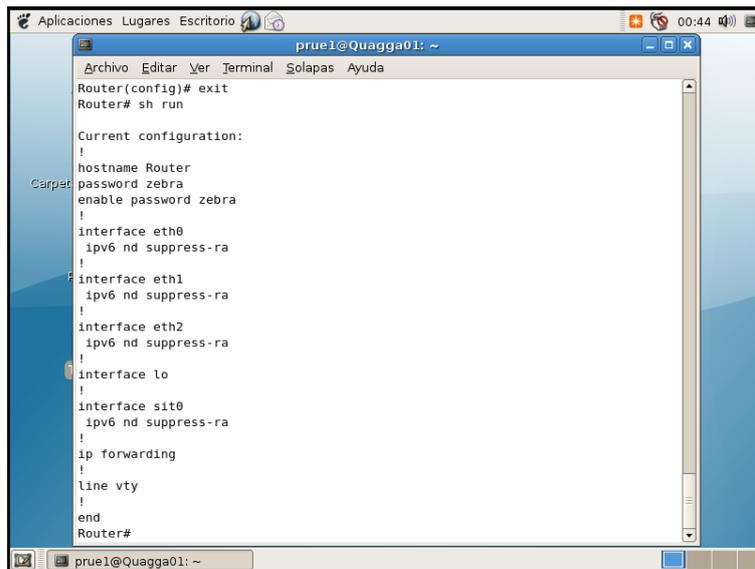
### Temporal

```
Debian:/# echo "1" > /proc/sys/net/ipv4/ip_forward
```

### Permanente

```
Debian:/# echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
```

Si no se pone ninguna de las dos opciones el servidor no trabajaría, para corroborar que está trabajando el **ipforwarding**, entramos al modo de súper usuario en el **zebra.conf** de Quagga, ya que estemos ahí le ponemos **shrun**, en la parte de abajo muestra **ipfordwaring**, debajo de la interfaz lo (*loopback*), si no muestra esta línea significa que no está activa.



```
prue1@Quagga01: ~  
Router(config)# exit  
Router# sh run  
Current configuration:  
!  
hostname Router  
password zebra  
enable password zebra  
!  
interface eth0  
  ipv6 nd suppress-ra  
!  
interface eth1  
  ipv6 nd suppress-ra  
!  
interface eth2  
  ipv6 nd suppress-ra  
!  
interface lo  
!  
interface sit0  
  ipv6 nd suppress-ra  
!  
ip forwarding  
!  
line vty  
!  
end  
Router#
```

Ilustración 12: Cómo queda la configuración al final.

## Habilitando el NAT en Debian

Por último utilizaremos el traductor de direcciones de red (**NAT** por sus siglas en inglés) para el Debian, con el fin de que las personas externas a nuestra red, que estén del otro lado de su red no puedan ver sus máquinas. Para que las máquinas de la Intranet puedan acceder al Internet deben de ser a través de una dirección pública, dado que se cuenta con segmentos de direcciones privadas, no podrían, **NAT** se encarga de hacer esa conversión, de una dirección privada a pública y viceversa cuando regresa el paquete.

Para lograr esto utilizaremos el comando **iptables**, con una serie de indicadores según sea la sintaxis del comando.

Aquí lo dividimos en 3 partes, la primera conformada por 4 líneas, de las cuales se refieren al vaciado de reglas, la segunda parte igual tiene 4 líneas de código, se refieren políticas por defecto a *ACCEPT*, y por último la tercera parte, a la que corresponde a los servicios de **Nateo** o **NAT**.

Las primeras 2 partes, no se modifican para nada, en cambio en la 3 tendrían que modificar el segmento de red que corresponda a su configuración, para esta proyecto queda así.

Primera parte, vaciado o limpieza de reglas, con estos 4 comandos se eliminan todas reglas que pudiera tener el sistema operativo.

```
Quagga:~# iptables -F
Quagga:~# iptables -X
Quagga:~# iptables -Z
Quagga:~# iptables -t nat -F
```

Segunda parte, servicio de *NAT*.

```
Quagga:~# iptables -t nat -A POSTROUTING -s 10.10.100.0/24 -o eth0 -j
MASQUERADE
Quagga:~# iptables -t nat -A POSTROUTING -s 10.10.200.0/24 -o eth0 -j
MASQUERADE
```

En caso de que tengan redes adentro se tendrían que agregar también, por ejemplo tenemos otro ruteador con la dirección 10.10.100.0/24 la cual se conecta directamente a nuestra interfaz, y otra dirección que es la 10.10.50.0/24, en este caso como la agregamos la 100 tendríamos que agregar la regla con el segmento de la 50 quedando así:

```
Quagga:~# iptables -t nat -A POSTROUTING -s 10.10.50.0/24 -o eth0 -j
MASQUERADE
```

## Verificando la configuración del servidor

Verificar las *iptables* que se agregaron

```
iptables -t nat -L -n
```

Para verificar la tabla del NAT de las reglas anterior mente se introdujeron.

```
iptables -L -n
```

Para verificar la reglas insertadas anterior mente.

Un programa para monitorear la interfaz en específico es el iptraf, el cual se instala de de la siguiente forma.

```
apt-getinstalliptraf
```

Con esto se termina de instalar y configurar por completo el Quagga, quedando de esta forma la configuración del Quagga, esto se obtiene con el comando *shrun*, dentro de las configuraciones del Quagga.

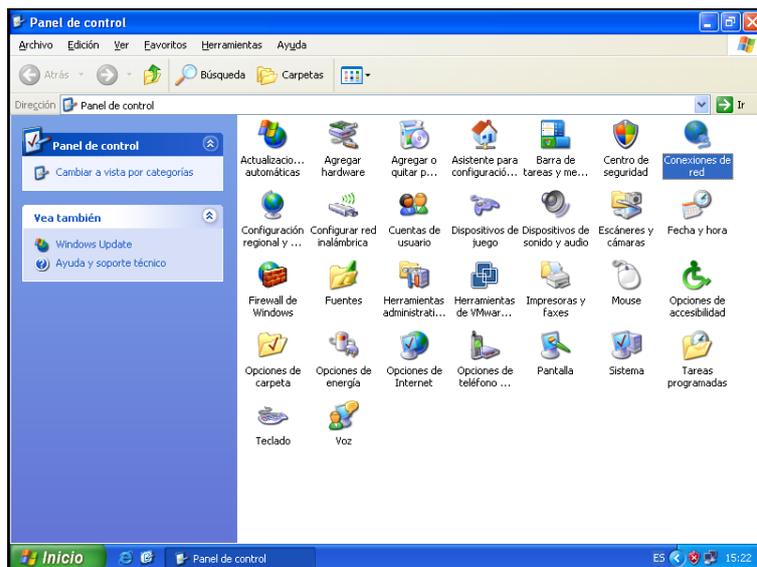
## Configuración de las máquinas para la red

Lo que sigue es sencillo, sólo es configurar las máquinas que vaya a tener la red, según sea el caso. Se presenta, a modo de ejemplo, la configuración de una máquina con Windows XP.



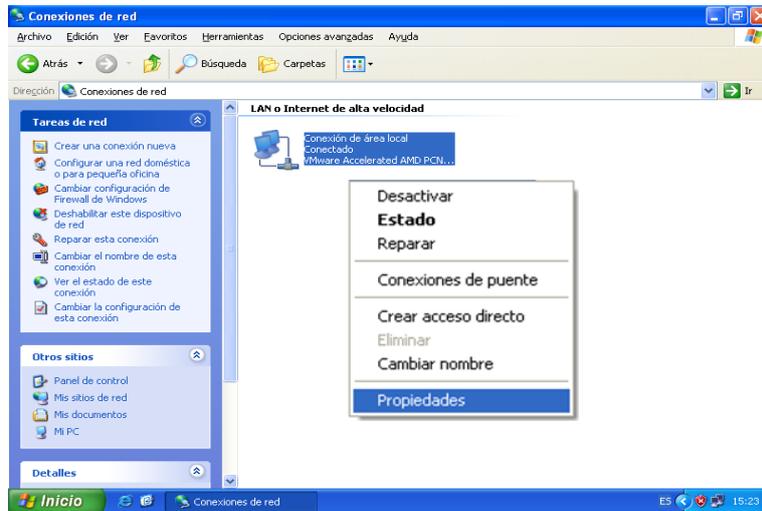
**Ilustración 12: Menú inicio del Windows.**

Estando en la máquina, le damos en Inicio, Panel de control.



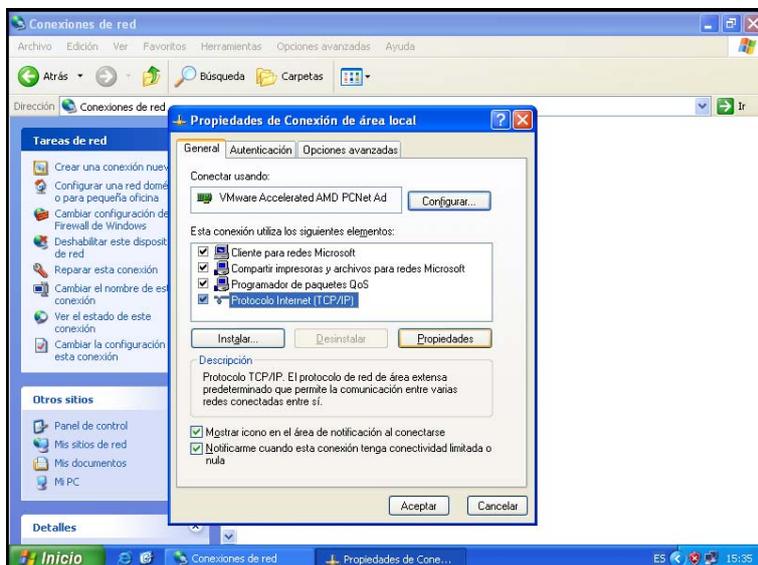
**Ilustración 13: Panel de control.**

Ya que abre panel de control, abrimos Conexiones de red.



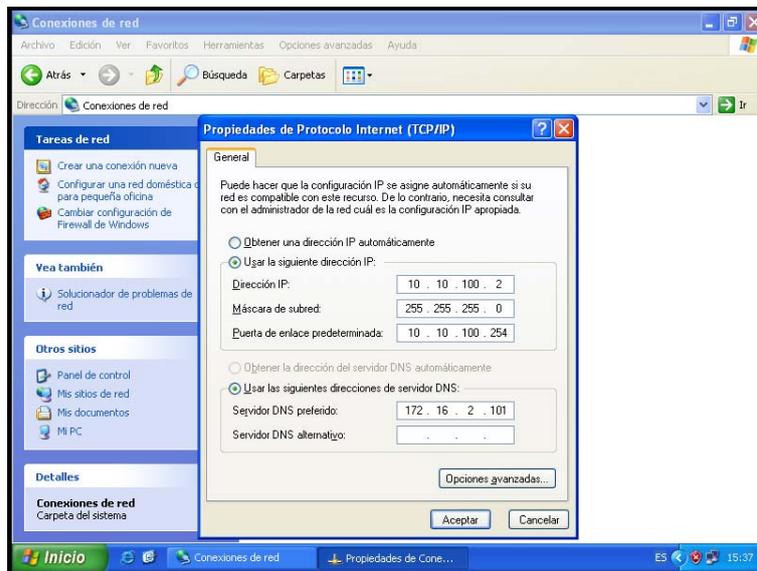
**Ilustración 14: Propiedades de la interfaz de red.**

Seleccionamos conexión de área de red, y con el segundo botón del ratón le damos arriba del icono, con esto desplegara un pequeño menú, en el cual tenemos que seleccionar propiedades.



**Ilustración 15: Menú de configuración de la tarjeta de red.**

Ya que les muestra esta ventana ahí que hacer dos cosas, la primera buscar y seleccionar la línea donde dice Protocolo de Internet [TCP/IP], la segunda darle en el botón propiedades.



**Ilustración 16: Configurar la IP a la máquina.**

Se abrirá una nueva ventana, en esta pondremos la IP, máscara, puerta de enlace y Servidor *DNS* de nuestra red.

Ya que se pusieron le damos en Aceptar a las 2 ventanas y cerramos la ventana de conexiones de red; con esta configuración hemos terminado con la configuración de la máquina, y por lo tanto debe de tener acceso a Internet.

### **Comparación**

En esta parte del proyecto se realizaron varias pruebas. Las pruebas se realizaron entre 1 ruteador y 2 máquinas con Quagga, utilizando el programa D-ITGel cual es un inyector de paquetes; dicho programa se instalara en la máquina Acer para él envió y en la DC7600 como receptor de paquetes, con este programaveremos el desempeño del envío de información de los 3 equipos.

Las primeras pruebas se realizaron con una máquina es una COMPAQ modelo Evo, adquirida en el 2001 por la universidad, una tarjeta de red integrada a 100 Mbps y una de PCI a la misma velocidad, 80 GB en disco duro 5400 RPM, con 1GB en *DIMM* para la memoria *RAM* y 2.0 GHz en procesador Pentium IV.

La segunda ronda se realizó con una máquina ensamblada, con 2GB en *DDR3* en *RAM*, un procesador i7 a 2.66 GHz, disco duro de 320 GB a 7200 RPM, y 3 tarjetas de red, 1 integrada a 1 GHz y las otras 2 de PCI a 100 Mbps.

Y por último tenemos un router de la marca Cisco System modelo 2800, con 2 puertos integrados a 100 Mbps cada uno.

El programa a utilizar para inyectar los paquetes se llama D-ITG el cual se descarga del siguiente link o liga:

["http://www.grid.unina.it/software/ITG/download.php"](http://www.grid.unina.it/software/ITG/download.php), en este caso bajamos la versión *D-ITG-2.6.1* de GNU/Linux and Windows sourcecode (*zip format*) con él se podrá realizar las pruebas, pero para mayor comodidad existe un programa de complemento que nos permitirá utilizar el D-ITG de manera gráfica, el cual se descarga de la página ["http://www.semken.com/projekte/index.html"](http://www.semken.com/projekte/index.html).

El programa se eligió basándose en una tesis realizada en la Universidad de Quintana Roo, por el *Ing. Luis Jesús Oliva Canché*, denominada *DISEÑO DE UNA ARQUITECTURA PARA MEDICIÓN DE ANCHO DE BANDA EN AMBIENTES CONTROLADOS*, al igual de una serie de pruebas realizadas en la *Escuela de Computación de la Universidad Central de Venezuela* en la cual se probaron 8 programas de inyección de paquetes, se probaron diversos factores en las pruebas realizadas, como se muestra en la tabla 1, el documento de la prueba se puede descargar de la siguiente liga ["www.ciens.ucv.ve/escueladecomputacion/documentos/archivo/86"](http://www.ciens.ucv.ve/escueladecomputacion/documentos/archivo/86)

**Tabla 1: Características Principales de las Herramientas Evaluadas.**

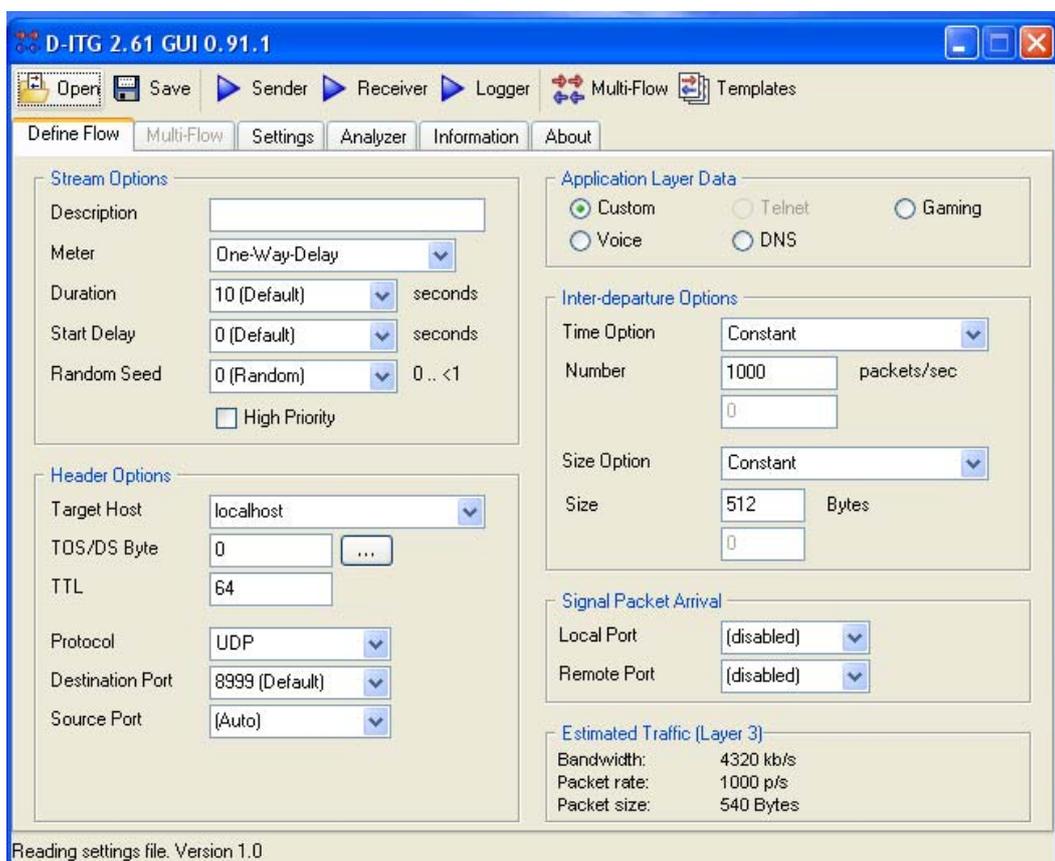
Aspecto	Netperf	D-ITG	NetStress	MGEN	LANforge	Network Traffic Generator	Rude & Crude	WlanTV
Versión evaluada	2.4.4	2.6.1d	1.0.8350	4.2b6	5.0.9	0.1.3.1	0.62	1.3.0
Última actualiz.	10/2007	09/2008	09/2008	10/2005	04/2009	01/2003	09/2002	12/2008
Gratis	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí
Open source	Sí	Sí	No	Sí	No	Sí	Sí	Sí
A/P	Activa	Activa	Activa	Activa	Activa	Activa	Activa	Pasiva
Privilegios	Usuario	Usuario	Usuario	Usuario	Usuario	Root	Root	Root
Plataforma soportada	Linux	Linux, Windows	Windows	Linux, Windows	Linux, Windows	Linux	Linux	Linux, Windows
Protocolo de red	IPv4, IPv6	IPv4, IPv6, ICMPv4, ICMPv6	IPv4	IPv4, IPv6	IPv4, IPv6, ICMPv4, ICMPv6	IPv4	IPv4	IPv4, IPv6
Protocolo de transporte	TCP, UDP	TCP, UDP	TCP	UDP	TCP, UDP	TCP, UDP	UDP	TCP, UDP
Resultados reportados	Throughput	Retardo, jitter y throughput	Throughput	Retardo y throughput	Retardo, jitter y throughput	--	Retardo, jitter y throughput	Throughput
Interfaz de usuario	Consola	Consola	GUI	Consola	GUI	Consola	Consola	GUI
Sincroniz. requerida	No	No	No	No	No	--	Sí	--

Según las necesidades de proyecto se eligió el programa D-ITG, por tener interfaz gráfica, ser gratuito, protocolos a utilizar, ser uno de los más recientes en su versión, entre otras cosas.

El inyector de paquetes consta de varios módulos, los cuales son: *ITGSend*, *ITGRecv*, *ITGDec*, *ITGLog*, *ITGManager*, cada uno de estos módulos tiene su función, las cuales se explican en el manual que viene integrado en el programa cuando se descarga, para poder utilizarlo se tiene que acceder desde MS-DOS, mediante comandos, por eso se optó por el programa de complemento, que integra todos los módulos del D-ITG y los junta en una sola aplicación gráfica, lo que lo hace más fácil de utilizar y de interpretar los datos.

Para lograr esto se tiene que descargar el *D-ITG GUI 0.91.1 beta*, el cual se tiene que descomprimir en la misma carpeta donde se encuentran los archivos

del D-ITG, una vez que se tiene todos los archivos en la misma ubicación se tiene que verificar que en la máquina se tenga instalado el programa de Java de preferencia una versión reciente, mínimo la 6. Se cuenta con 2 opciones para empezar a correr programa de forma gráfica, es dándole doble click en el archivo *ITGGUI.jar* o entrar en una ventana de MS-DOS y poner `javaw -jar ITGGUI.jar`, tomando en cuenta que tenemos que estar dentro de la carpeta donde se localiza el archivo *ApplicationLayer Data*.



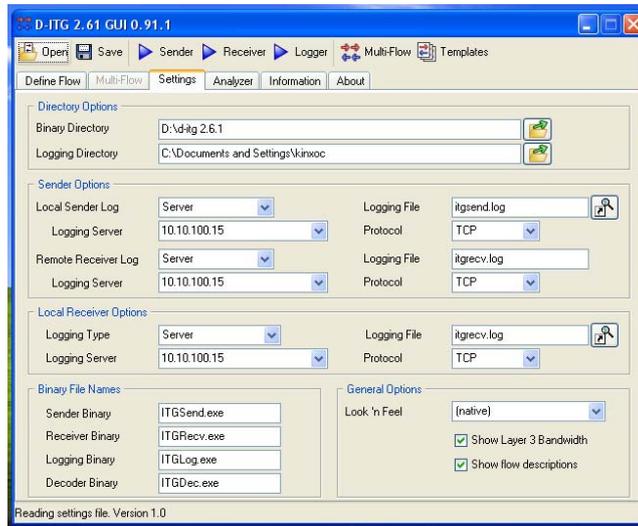
**Ilustración 17: Programa alterno para modalidad grafica del D-ITG.**

Una vez que está abierta la ventana, se puede empezar a trabajar con el programa, en la primera pestaña notamos que está dividido en 6 secciones, de las cuales solo se usaron 4 y la 6ta como información, los cuales se explicarán brevemente cada una de ellas.

La primera sección se llama *StreamOptions* y consta de 5 opciones, de las cuales sólo ocuparemos la de la duración, en este caso 60 segundos, la segunda parte se llama *HeaderOptions* y consta de 6 opciones las cuales están divididas en 2, las primeras 3 que sólo usamos la primera opción *Target Host* donde ponemos la dirección a donde se mandarán los paquetes, y en las segundas 3 se especifica qué tipo de protocolo, el puerto destino y el puerto de salida, la tercera sección es la *ApplicationLayer Data* en la cual se especifica qué tipo de información se enviará, la cuarta sección es *Inter-departureOptions* en esta parte consta de 6 opciones las cuales se activan cuando está seleccionada la opción de *Custom* en la sección anterior, en esta parte sólo usaremos 2 partes, la de *Number* en la que ponemos la cantidad de paquetes que queremos que se transmitan por segundo y la de *Size* con la que indicamos que tamaño tendrá cada paquete que se transmitirá. Por último, tenemos la sección de información en la que encontraremos 3 datos del envío, el primero el ancho de banda para el envío, después la velocidad de los paquetes que se refiere a cuantos paquetes por segundo y la tercera el tamaño real del paquete que se enviará, si ponemos que el paquete mida 512 Bytes nos mostrará 540 Bytes, por la información que se agrega al paquete para su envío.

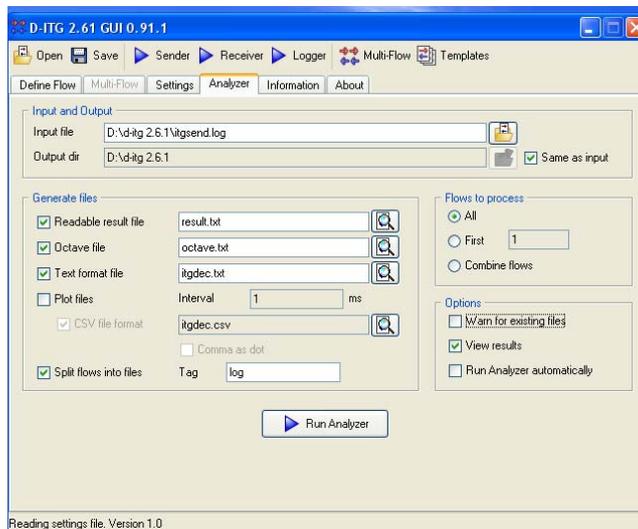
A continuación tenemos las 3 opciones de *Sender*, *Receiver* y *Logger*, con las cuales se podrá hacer la inyección de paquetes, primero se tiene que activar la opción de *Logger* para crear los archivos de registro; seguido se activa *Receiver*, en esta opción se prepara el programa para empezar a recibir los paquetes que le lleguen; y por último la opción de *Sender* en donde empezará el programa a mandar los paquetes como se configuró anteriormente.

En la pestaña de *Settings* podemos modificar a donde se enviarán los paquetes, ya sea local o remotamente a un servidor, y en que archivo se almacenará la información del envío de los paquetes.



**Ilustración 18: La pestaña de Settings.**

En la pestaña de *Analyzer*, se podrá crear el archivo con datos del envío como: cuantos paquetes llegaron, si tuvieron errores, total del ancho de banda, entre otros. Sólo presionamos el botón *RunAnalyzer*.



**Ilustración 19: La pestaña de analyzer.**

# Capítulo 4

---

## **Pruebas a realizar**

En este último paso se realizarán varias pruebas, haciendo las mismas en cada uno de los tres equipos: 2 máquinas de escritorio, con el OS Debian y el programa de Quagga, y un ruteador de marca Cisco.

La prueba consiste en inyectar paquetes a los equipos, empezando con pocos paquetes e ir incrementando la cantidad por segundo, al igual el tamaño del paquete, con el fin de ver qué equipo puede soportar más paquetes por segundo y comparar los resultados de las pruebas, analizando si tiene un desempeño similar entre los 3 equipos.

Las pruebas se realizarán primero al equipo COMPAQ modelo EVO, seguidamente a la máquina ensamblada y por último al ruteador Cisco. Cada equipo recibirá 8 pruebas, las cuales se dividen en 2 partes: la primera parte consta de 5 pruebas y la segunda parte de 3.

La primera parte está dividida en 5 pruebas, en la opción de *Define Flow* un tiempo de 60 segundos y mediante el protocolo TCP, las cuales se dividen de la siguiente forma:

- 1.- Será la que está definida, número de 1000 P/s (Paquetes por segundo) y un tamaño de 512 Bytes cada paquete.
- 2.- Se incrementará a 100,000 P/s y un tamaño de 1024 Bytes
- 3.- Será a 1, 000,000 P/s y un tamaño de 2048 Bytes
- 4.- Al máximo de envío por segundo que es a 9, 999,999 P/s con un tamaño de 4096 Bytes
- 5.- Seguirá a 9, 999,999 P/s pero con un tamaño de 8192 Bytes.

La segunda parte está conformada por 3 pruebas en la modalidad de *Multi-Flow*, en la cual permite enviar distintos protocolos y varios flujos de paquetes al mismo tiempo. Las 3 pruebas se dividen de la siguiente manera:

6.- Se enviarán 2 flujos, con un número de 9, 999,999 P/s a 8192 Bytes, como quien dice el doble de paquetes que en la quinta prueba.

7.- Con la misma cantidad de P/s y tamaño con la diferencia que serán 4 flujos a la vez

8.- En la última prueba se pondrá a 8 líneas de flujo a la vez con el mismo número y tamaño que en las anteriores.

Para poder realizar estas pruebas se utilizarán 2 máquinas extras, una se utilizará para el envío de los paquetes y la otra que los recibe, pasando a través de las máquinas con Quagga, o el ruteador.

Con cada prueba que se realice el programa D-ITG crea archivos de registro, los cuales utilizaremos como referencia para realizar parte de las comparativas entre los 3 equipos.

En las máquinas con el Quagga se utilizará un programa llamado *iptraf*, el cual se instaló y permitirá ver en tiempo real el ancho de banda, y la cantidad de paquetes transmitidos por interfaz, permitiendo ver el desempeño de la máquina, sacando una imagen al segundo 59 de la prueba antes de que el programa deje de enviar paquetes y termine con la prueba.

### **Análisis de las pruebas realizadas**

En esta sección del proyecto analizaremos los resultados de las 8 pruebas realizadas, en los 3 equipos utilizados, mediante los resultados arrojados con el programa *Iptraf* en el Debian y el D-ITG con sus registros, los cuales generan un archivo TXT con el concentrado total de las pruebas; a continuación se mostrarán 2 ejemplos de los archivos generados.

### Ejemplo 1: primer prueba realizada.

```
-----  
Flownumber: 1  
From 192.168.8.5:2131  
To 10.10.80.15:8999  
-----  
Total time = 59.999000 s  
Total packets = 60000  
Minimum delay = 0.000000 s  
Maximum delay = 0.000000 s  
Average delay = 0.000000 s  
Average jitter = 0.000000 s  
Delay standard deviation = 0.000000 s  
Bytes received = 30720000  
Average bitrate = 4096.068268 Kbit/s  
Average packet rate = 1000.016667 pkt/s  
Packets dropped = 0 (0.00 %)  
-----
```

\*\*\*\*\* TOTAL RESULTS \*\*\*\*\*

```
-----  
Number of flows = 1  
Total time = 59.999000 s  
Total packets = 60000  
Minimum delay = 0.000000 s  
Maximum delay = 0.000000 s  
Average delay = 0.000000 s  
Average jitter = 0.000000 s  
Delay standard deviation = 0.000000 s  
Bytes received = 30720000  
Average bitrate = 4096.068268 Kbit/s  
Average packet rate = 1000.016667 pkt/s  
Packetsdropped = 0 (0.00 %)  
Error lines = 0  
-----
```

### Ejemplo 2: octava prueba realizada.

```
-----  
Flow number: 2  
From 192.168.8.5:2157  
To 10.10.80.15:9002  
-----  
Total time = 59.999000 s  
Total packets = 43315  
Minimum delay = 0.000000 s  
Maximum delay = 0.000000 s  
Average delay = 0.000000 s  
Average jitter = 0.000000 s  
Delay standard deviation = 0.000000 s  
Bytes received = 354836480  
-----
```

```

Average bitrate      = 47312.319205 Kbit/s
Average packet rate  = 721.928699 pkt/s
Packets dropped      = 0 (0.00 %)

```

```

-----
Flow number: 1
From 192.168.8.5:2158
To 10.10.80.15:8999

```

```

-----
Total time          = 59.999000 s
Total packets       = 41502
Minimum delay       = 0.000000 s
Maximum delay       = 0.000000 s
Average delay       = 0.000000 s
Average jitter      = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received      = 339984384
Average bitrate     = 45332.006733 Kbit/s
Average packet rate = 691.711529 pkt/s
Packets dropped     = 0 (0.00 %)

```

```

-----
***** TOTAL RESULTS *****

```

```

-----
Number of flows     = 2
Total time          = 60.015000 s
Total packets    = 84817
Minimum delay       = 0.000000 s
Maximum delay       = 0.000000 s
Average delay       = 0.000000 s
Average jitter      = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received    = 694820864
Average bitrate   = 92619.626960 Kbit/s
Average packet rate = 1413.263351 pkt/s
Packetsdropped     = 0 (0.00 %)
Error lines         = 0

```

Las líneas que están resaltadas en negro son los valores que se utilizarán para hacer las comparaciones.

Las pruebas realizadas con los 3 equipos arrojaron distintos resultados, ahora se mostrarán los resultados para cada máquina.

Ya que se tengan los 24 resultados en total, se analizarán la primera de cada equipo para hacer la comparación, así sucesivamente con las 8, una vez que se tengan separadas como antes se mencionó se mostrarán los resultados en 4

categorías: la primera será *Kbit/s*, la segunda *paquetes por segundo (pkt/s)*, la tercera *Bytes recibidos*, y por ultimo *Total de pkt*.

En los 4 análisis se mostrarán las tablas con las celdas de colores las cuales representande la siguiente manera:

Más paquetes enviados
2do lugar
3er lugar

Y también se mostrará una gráfica con los valores de los resultados obtenidos.

A continuación procederemos al análisis de los resultados.

### **Análisis mediante D-ITG**

En la primera prueba vemos que los 3 equipos tienen el mismo resultado, indicando que los 3 pasaron la misma cantidad de tráfico sin problemas. Pero a partir de la segunda prueba en adelante los resultados son distintos, indicando el comportamiento real de cada equipo para un tráfico diferente.

La máquina EVO ganó en las pruebas 3, 4, 5, indicando que a poco tráfico es la mejor opción. El ruteador 2800 ganó en la 2da prueba y la máquina ensamblada ganó en las pruebas 6, 7, 8, mostrando un mejor desempeño para una carga más pesada en el tráfico de paquetes.

**Tabla 2: Kbit/s.**

Datos Kbit/s	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Máquina EVO	4096.068268	79894.90385	85969.10002	90559.15412
Ensamblada	4096.068268	78792.37881	85894.55158	89424.12214
ruteador 2800	4096.068268	79967.50825	85852.22554	90317.75916
	Prueba 5	Prueba 6	Prueba 7	Prueba 8
Máquina EVO	92434.60724	92557.39351	92679.27906	92761.8551
Ensamblada	91467.93513	92619.62696	92739.35472	92859.22885
ruteador 2800	92194.30457	92408.39241	92701.97914	92791.97827

Figura 2: Kbit/s.

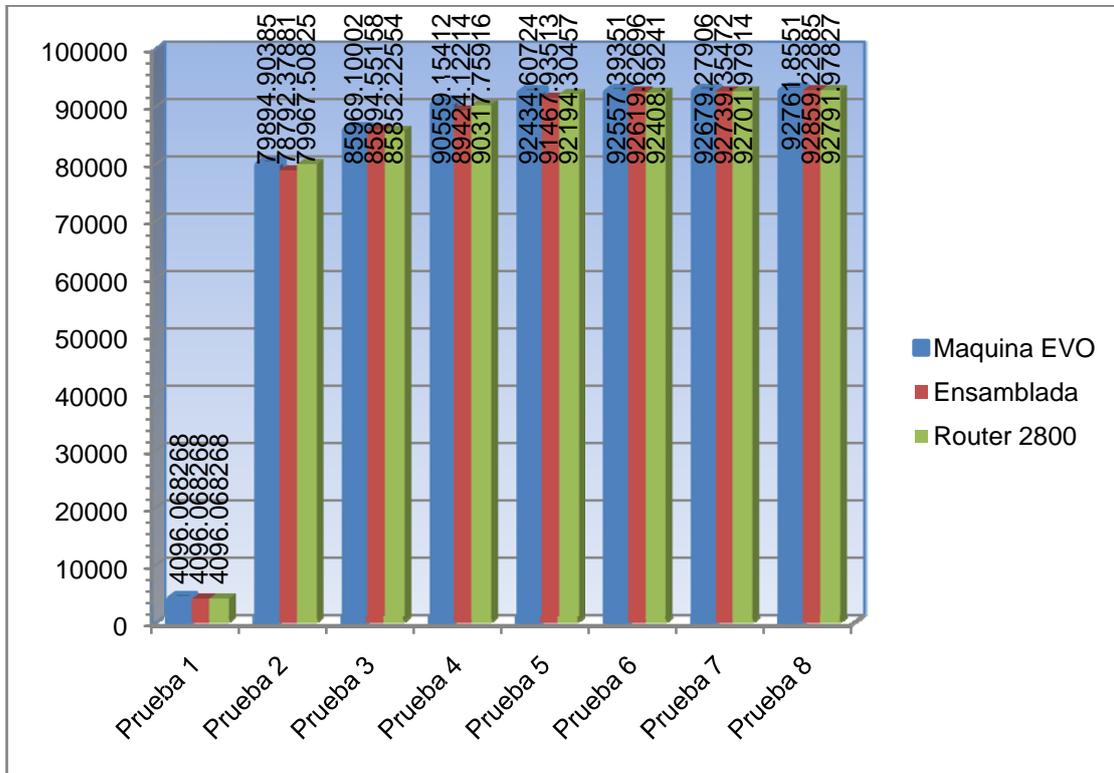


Tabla 3: pkt/s.

Datos pkt/s	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Máquina EVO	1000.016667	9752.79588	5247.137452	2763.646061
Ensamblada	1000.016667	9618.210304	5242.587376	2729.007634
ruteador 2800	1000.016667	9761.658722	5240.004	2756.279271
	Prueba 5	Prueba 6	Prueba 7	Prueba 8
Máquina EVO	1410.440174	1412.313744	1414.17357	1415.43358
Ensamblada	1395.689928	1413.263351	1415.090252	1416.919385
ruteador 2800	1406.773446	1410.040167	1414.519945	1415.893223

Figura 3: pkt/s.

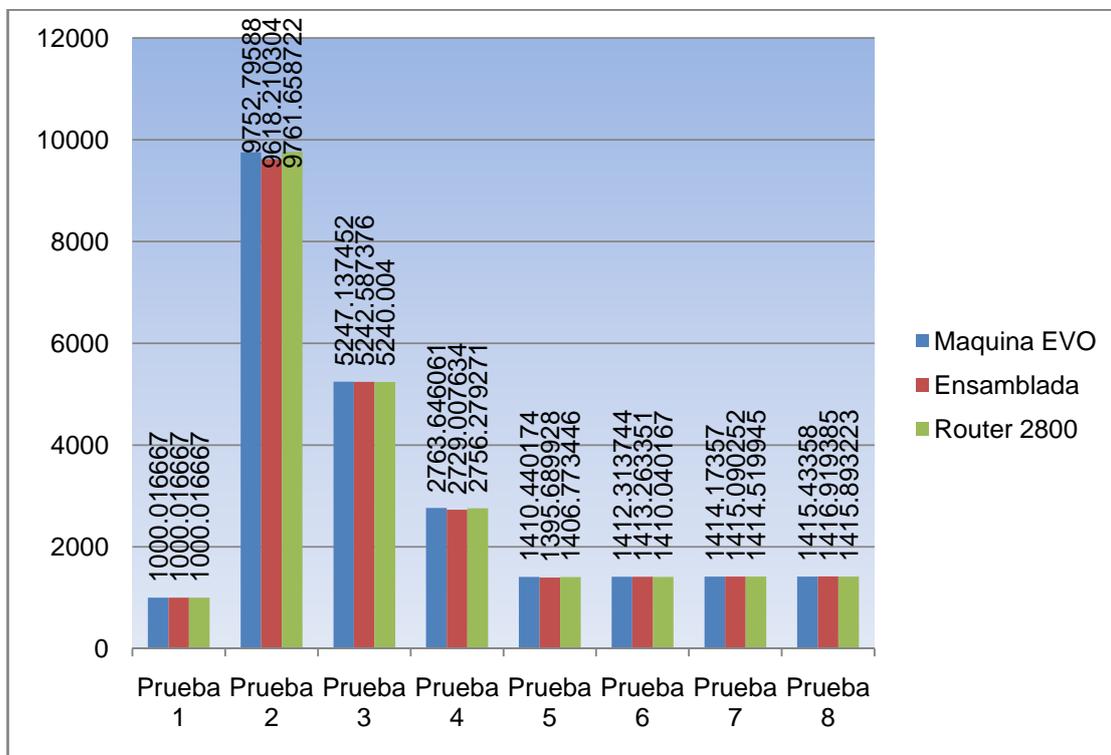


Tabla 4: Bytes recibidos.

Bytes recibidos	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Máquina EVO	30720000	599201792	644757504	679182336
Ensamblada	30720000	590932992	644198400	670658560
ruteador 2800	30720000	599736320	643880960	677371904
	Prueba 5	Prueba 6	Prueba 7	Prueba 8
Máquina EVO	693248000	694157312	695083008	695853056
Ensamblada	685998080	694820864	695533568	696606720
ruteador 2800	691445760	693051392	695427072	696090624

Figura 4: Bytes recibidos.

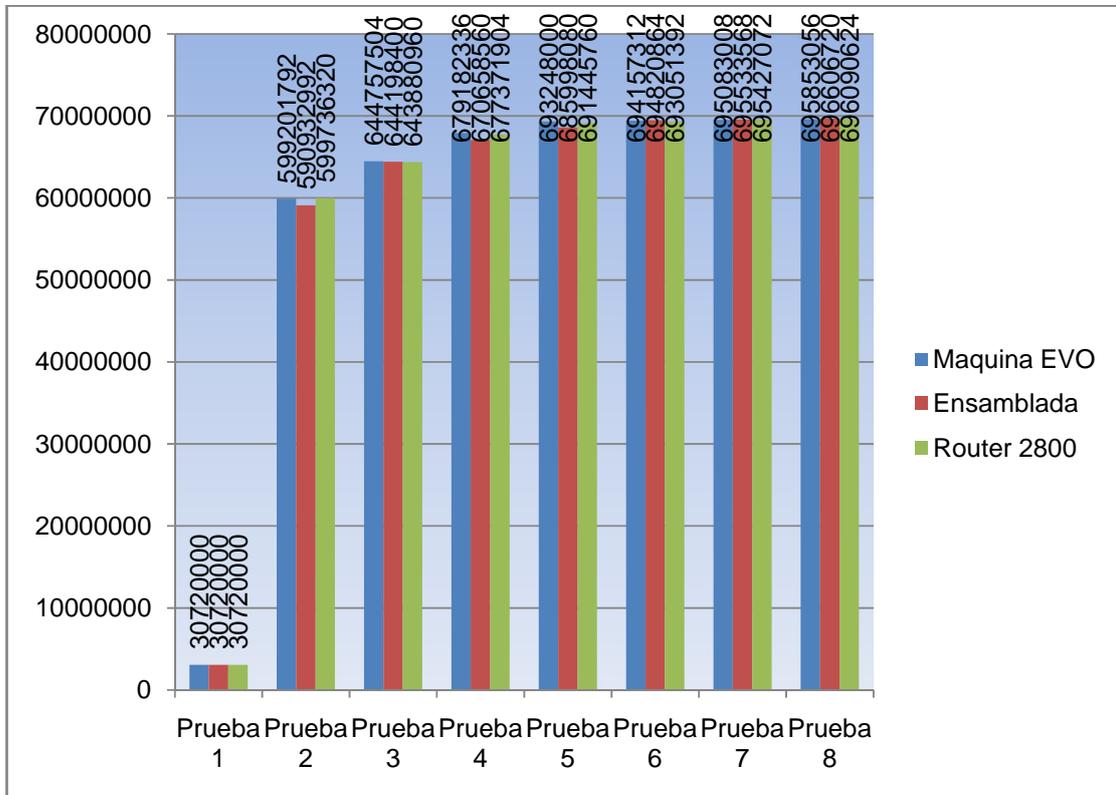
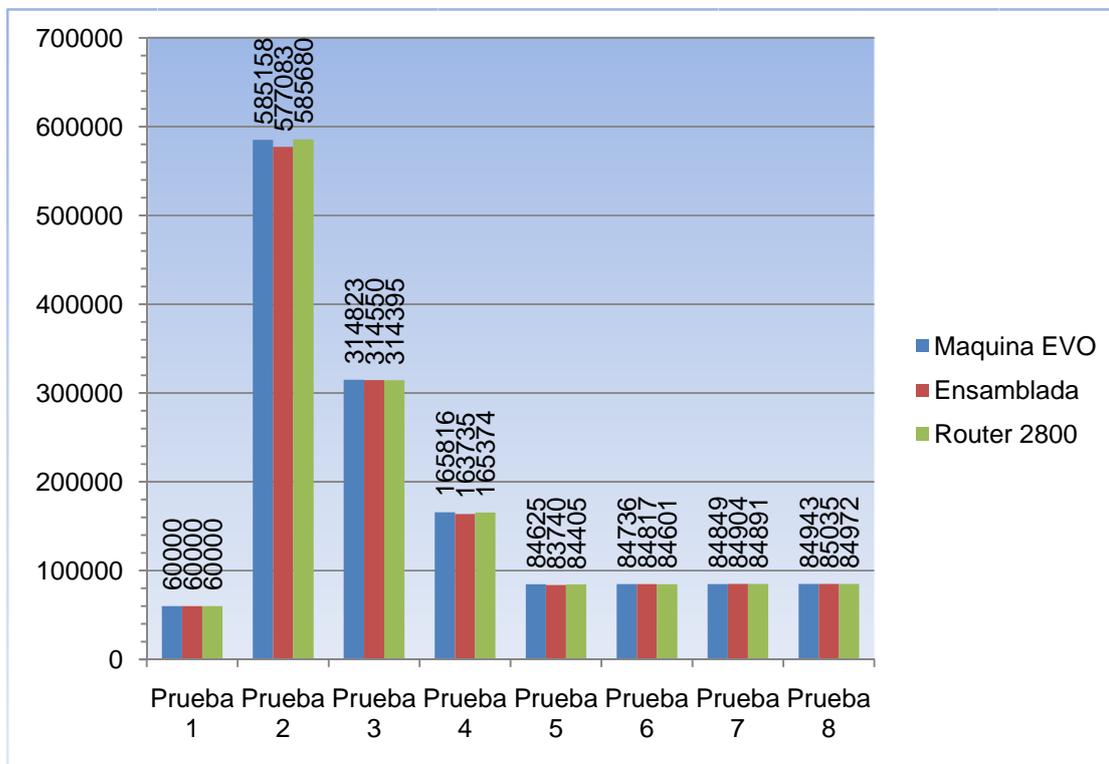


Tabla 5: Total de pkt.

Total de pkt	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Máquina EVO	60000	585158	314823	165816
Ensamblada	60000	577083	314550	163735
Ruteador 2800	60000	585680	314395	165374
	Prueba 5	Prueba 6	Prueba 7	Prueba 8
Máquina EVO	84625	84736	84849	84943
Ensamblada	83740	84817	84904	85035
Ruteador 2800	84405	84601	84891	84972

Figura 5: Total de pkt.



**Análisis con IPTraf**

La primera imagen en cada prueba corresponderá a la máquina Ensamblada y la segunda imagen a la máquina EVO, ambas con Quagga.

## Resultados de la primera Prueba.

```

IPTraf
-----
Total: Total: Incoming: Outgoing: Outgoing: Outgoing:
Packets Bytes Packets Bytes Packets Bytes
Total: 68979 4154806 58553 4811906 18426 95151
IP: 68979 4154806 58553 4811906 18426 95151
TCP: 68977 4154805 58553 4811906 18426 95151
UDP: 2 184 0 0 0 2
ICMP: 0 0 0 0 0 0
Other IP: 0 0 0 0 0 0
Non-IP: 0 0 0 0 0 0

Total rates: 5821.9 kbits/sec Broadcast packets: 0
            1188.4 packets/sec Broadcast bytes: 0

Incoming rates: 5685.9 kbits/sec
              885.4 packets/sec

Outgoing rates: 116.1 kbits/sec
              315.8 packets/sec

Elapsed time: 0:01
x-exit
    
```

```

IPTraf
-----
Total: Total: Incoming: Outgoing: Outgoing: Outgoing:
Packets Bytes Packets Bytes Packets Bytes
Total: 68984 42628151 18427 1185648 58557 4812713
IP: 68984 42628151 18427 1185648 58557 4812713
TCP: 68988 41549613 18425 957884 58555 4812569
UDP: 0 248 2 184 0 0
ICMP: 0 0 0 0 0 0
Other IP: 0 0 0 0 0 0
Non-IP: 0 0 0 0 0 0

Total rates: 5837.5 kbits/sec Broadcast packets: 0
            1379.6 packets/sec Broadcast bytes: 0

Incoming rates: 551.1 kbits/sec
              314.8 packets/sec

Outgoing rates: 5686.4 kbits/sec
              864.8 packets/sec

Elapsed time: 0:01
x-exit
    
```

```

IPTraf
-----
Interface Total In Out IP Non-IP Rate
eth0 68979 0 0 0 0 0.00 kbits/sec
eth1 68984 68979 68984 0 0 5794.80 kbits/sec
eth2 0 0 0 0 0 0.00 kbits/sec

Elapsed time: 0:01
Total, IP, Non-IP, and RateIP are packet counts.
Up/Down/Pkts/Bytes-scroll window x-exit
    
```

Ilustración 20: Prueba 1 IPTraf, Ensamblada.

```

IPTraf
-----
Total: Total: Incoming: Outgoing: Outgoing: Outgoing:
Packets Bytes Packets Bytes Packets Bytes
Total: 571137 5547996 383241 5486436 188896 18157224
IP: 571137 5486888 383241 5292736 188896 17323764
TCP: 571134 5486888 383241 5292736 188893 17323764
UDP: 0 218 0 0 0 218
ICMP: 0 0 0 0 0 0
Other IP: 0 0 0 0 0 0
Non-IP: 0 184 0 0 0 0

Total rates: 75289.8 kbits/sec Broadcast packets: 0
            9671.8 packets/sec Broadcast bytes: 0

Incoming rates: 73880.2 kbits/sec
              6489.4 packets/sec

Outgoing rates: 1378.3 kbits/sec
              3372.4 packets/sec

Elapsed time: 0:01
x-exit
    
```

```

IPTraf
-----
Total: Total: Incoming: Outgoing: Outgoing: Outgoing:
Packets Bytes Packets Bytes Packets Bytes
Total: 584964 5758828 187998 11879908 398866 5638126
IP: 584964 5661746 187998 11879908 398866 5582546
TCP: 584935 5661746 187998 11879908 398863 5582546
UDP: 5 208 2 184 3 156
ICMP: 0 0 0 0 0 0
Other IP: 0 0 0 0 0 0
Non-IP: 0 0 0 0 0 0

Total rates: 78272.5 kbits/sec Broadcast packets: 0
            18189.8 packets/sec Broadcast bytes: 0

Incoming rates: 1812.8 kbits/sec
              1288.8 packets/sec

Outgoing rates: 78851.1 kbits/sec
              6749.8 packets/sec

Elapsed time: 0:01
x-exit
    
```

```

IPTraf
-----
Interface Total In Out IP Non-IP Rate
eth0 0 0 0 0 0 0.00 kbits/sec
eth1 178026 178026 0 0 24778.80 kbits/sec
eth2 171496 171496 0 0 22628.48 kbits/sec

Elapsed time: 0:01
Total, IP, Non-IP, and RateIP are packet counts.
Up/Down/Pkts/Bytes-scroll window x-exit
    
```

Ilustración 21: Prueba 1 IPTraf, EVO.

## Resultados de la segunda prueba

The screenshot shows two terminal windows displaying IPTraf statistics for the eth0 interface. The top window shows overall statistics, and the bottom window shows per-interface activity.

	Total Packets	Total Bytes	Incoming Packets	Incoming Bytes	Outgoing Packets	Outgoing Bytes
Total	941129	711029K	628778	684181K	312351	1688718K
IP:	941129	697852K	628778	685358K	312351	1248418K
TCP:	941127	697852K	628778	685358K	312349	1248408K
UDP:	2	184	0	0	2	184
ICMP:	0	0	0	0	0	0
Other IP:	0	0	0	0	0	0
Non-IP:	0	0	0	0	0	0

Interface	Total	IP	NonIP	BuiltIP	Activity
lo	0	0	0	0	0.00 kbits/sec
eth0	949998	948998	0	0	96813.60 kbits/sec
eth1	941006	941006	0	0	96274.00 kbits/sec
eth2	0	0	0	0	0.00 kbits/sec

Ilustración 22: Prueba 2 IPTraf, Ensamblada.

The screenshot shows two terminal windows displaying IPTraf statistics for the eth0 interface. The top window shows overall statistics, and the bottom window shows per-interface activity.

	Total Packets	Total Bytes	Incoming Packets	Incoming Bytes	Outgoing Packets	Outgoing Bytes
Total	586541	383760K	338595	374636K	167946	986919K
IP:	586541	376618K	338595	369895K	167946	671794K
TCP:	586539	376618K	338595	369895K	167944	671784K
UDP:	2	184	0	0	2	184
ICMP:	0	0	0	0	0	0
Other IP:	0	0	0	0	0	0
Non-IP:	0	0	0	0	0	0

Interface	Total	IP	NonIP	BuiltIP	Activity
lo	0	0	0	0	0.00 kbits/sec
eth0	282721	282721	0	0	23218.40 kbits/sec
eth1	282153	282153	0	0	28248.40 kbits/sec

Ilustración 23: Prueba 2 IPTraf, EVO.

### Tercera Prueba

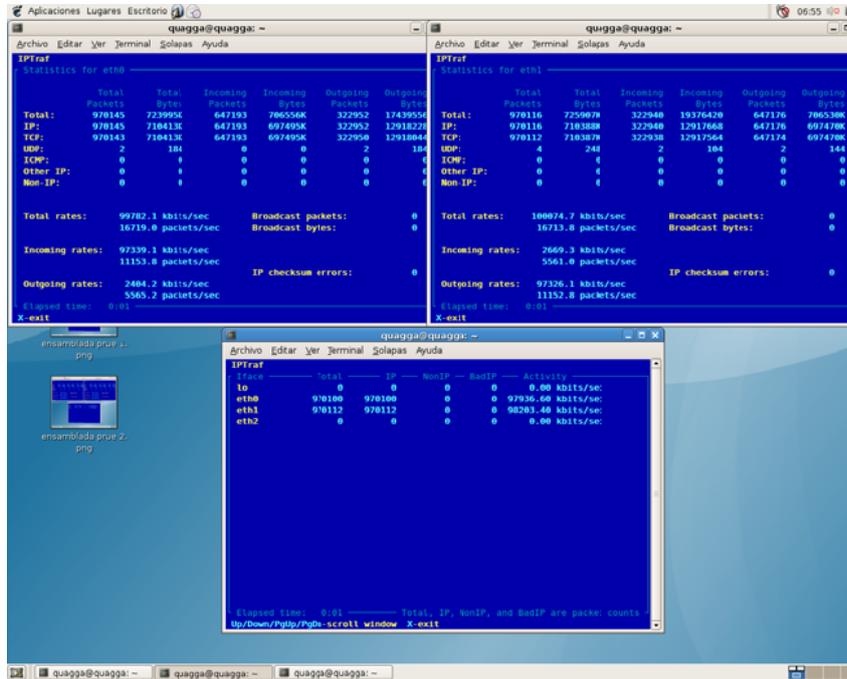


Ilustración 24: Prueba 3 IPTraf, Ensamblada.

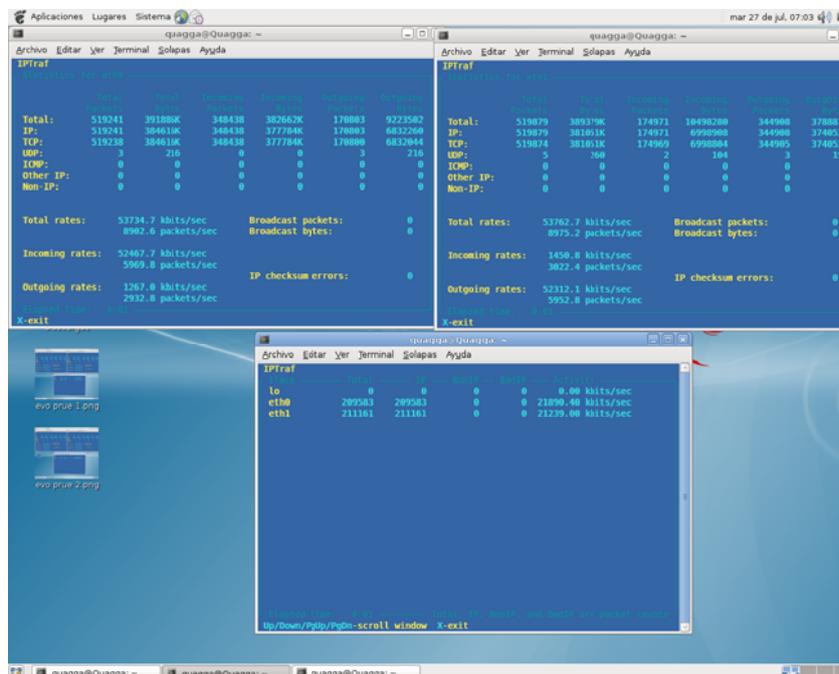


Ilustración 25: Prueba 3 IPTraf, EVO.

### Cuarta Prueba

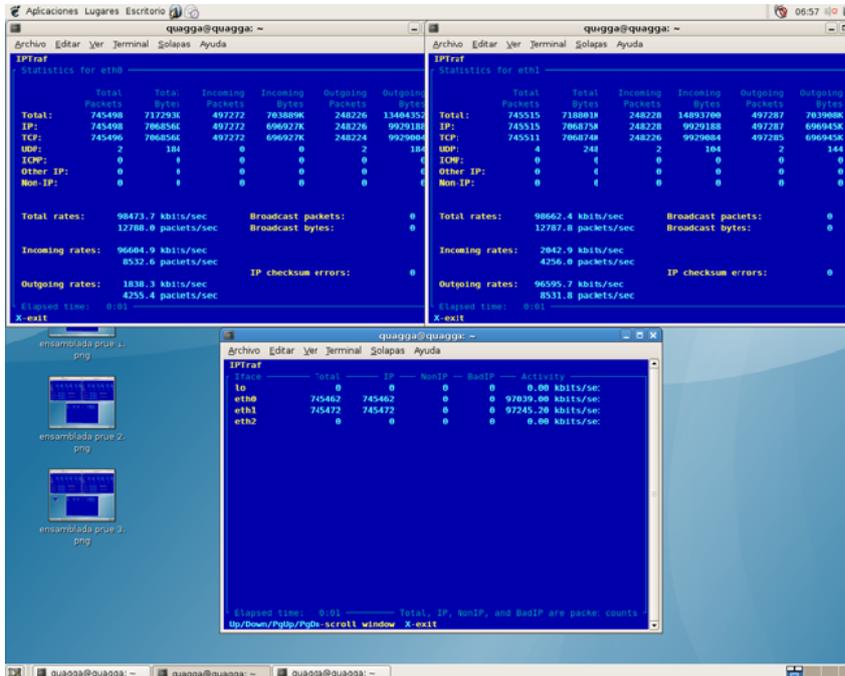


Ilustración 26: Prueba 4 IPTraf, Ensamblada.

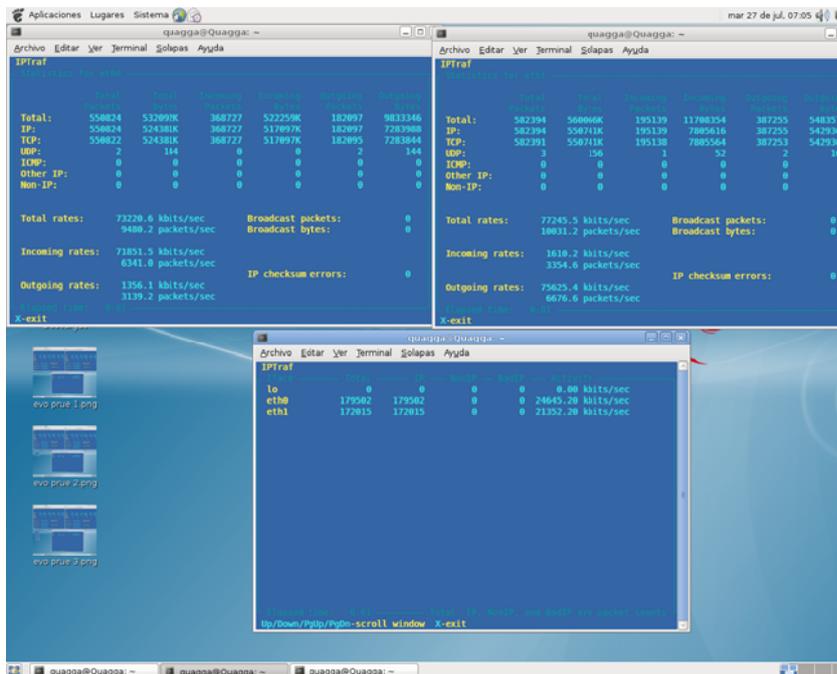


Ilustración 27: Prueba 3 IPTraf, EVO.

Quinta prueba

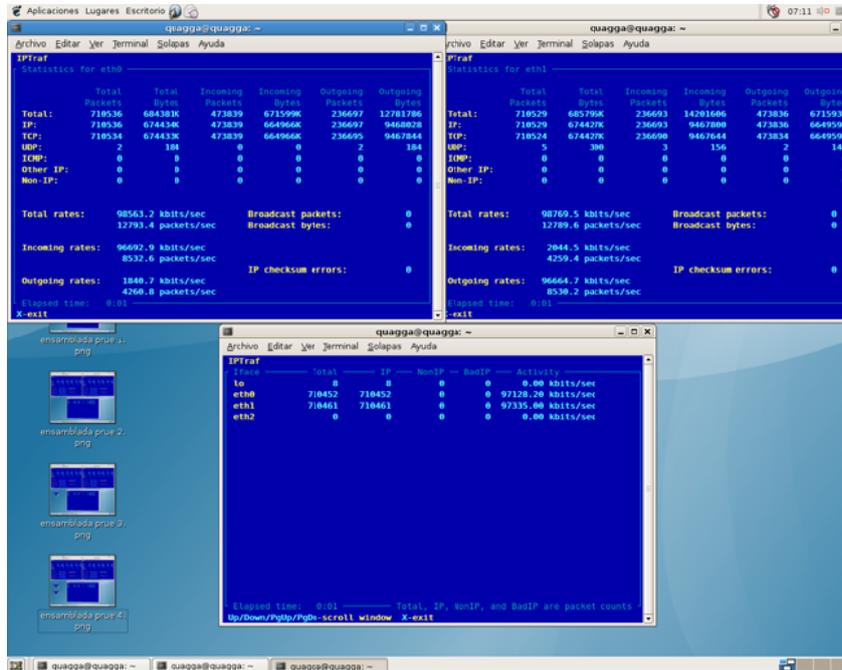


Ilustración 28: Prueba 5 IPTraf, Ensamblada.

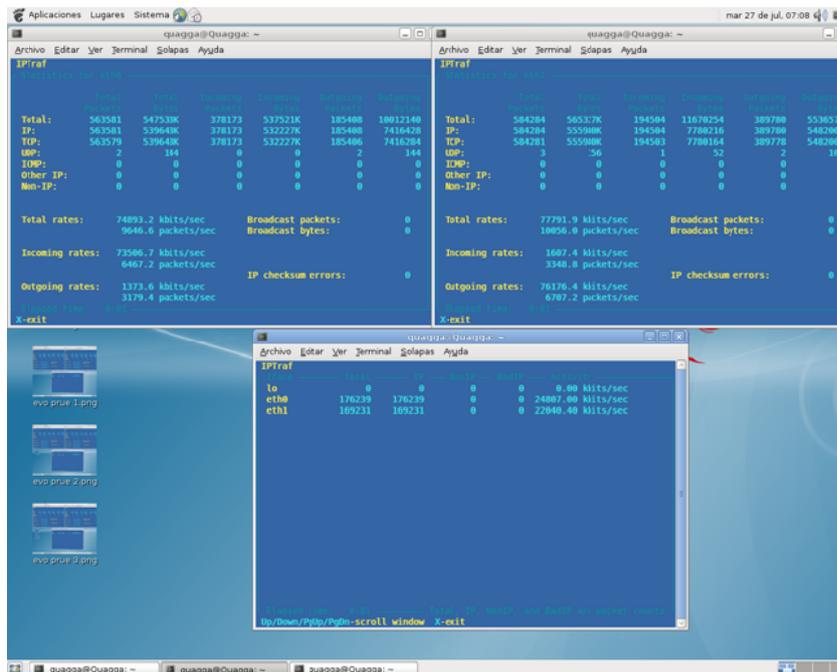


Ilustración 29: Prueba 5 IPTraf, EVO.

Sexta prueba

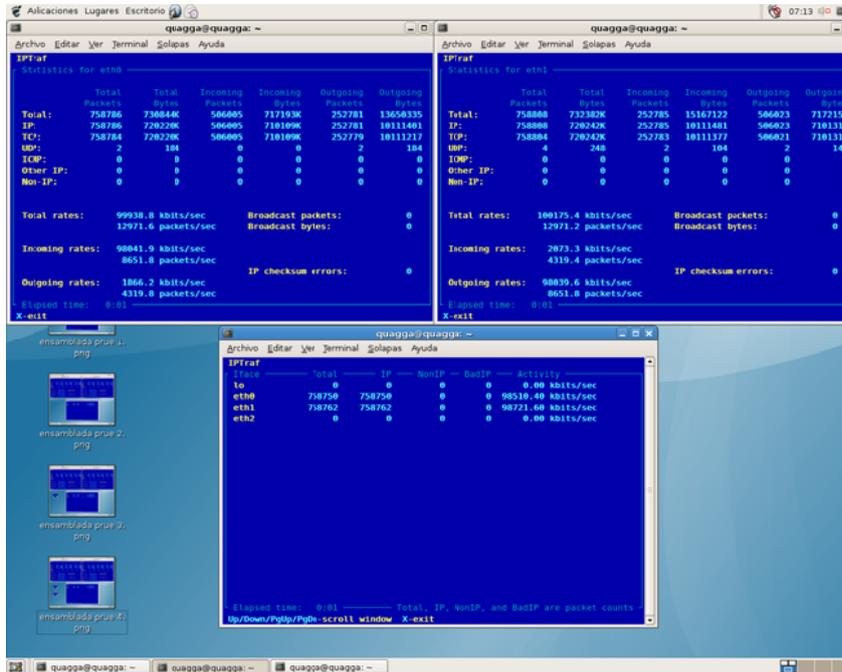


Ilustración 30: Prueba 6 IPTraf, Ensamblada.

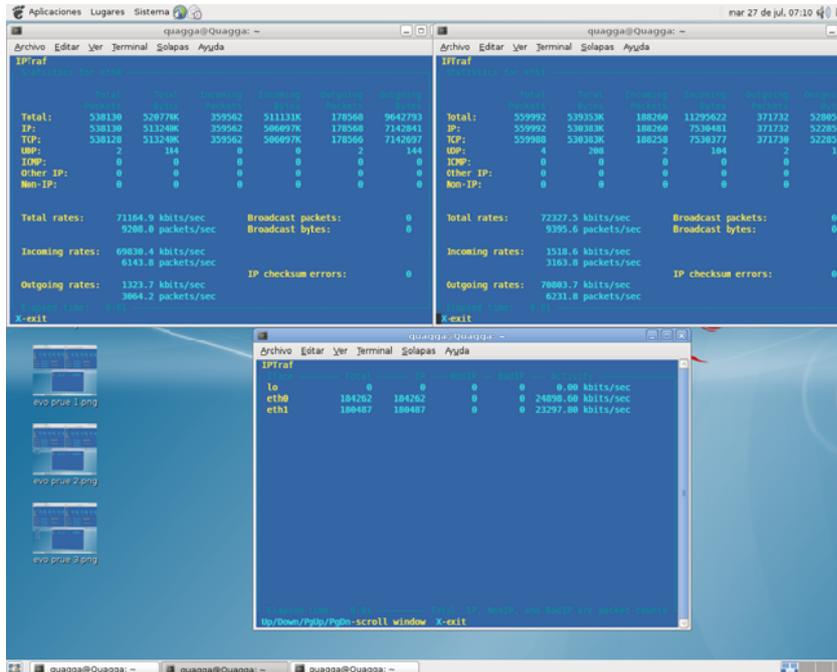


Ilustración 31: Prueba 6 IPTraf, EVO.

Séptima prueba

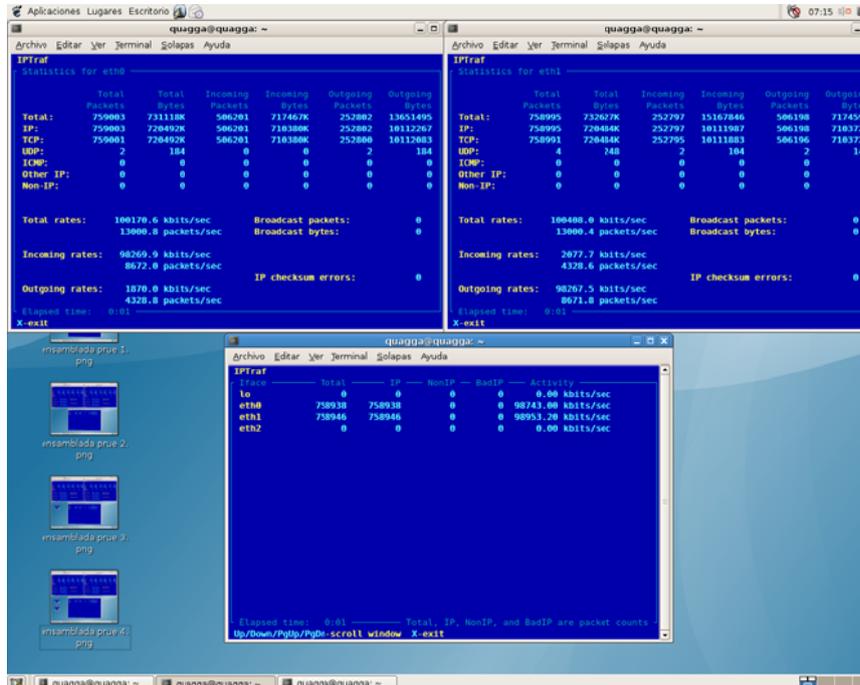


Ilustración 32: Prueba 7 IPTraf, Ensamblada.

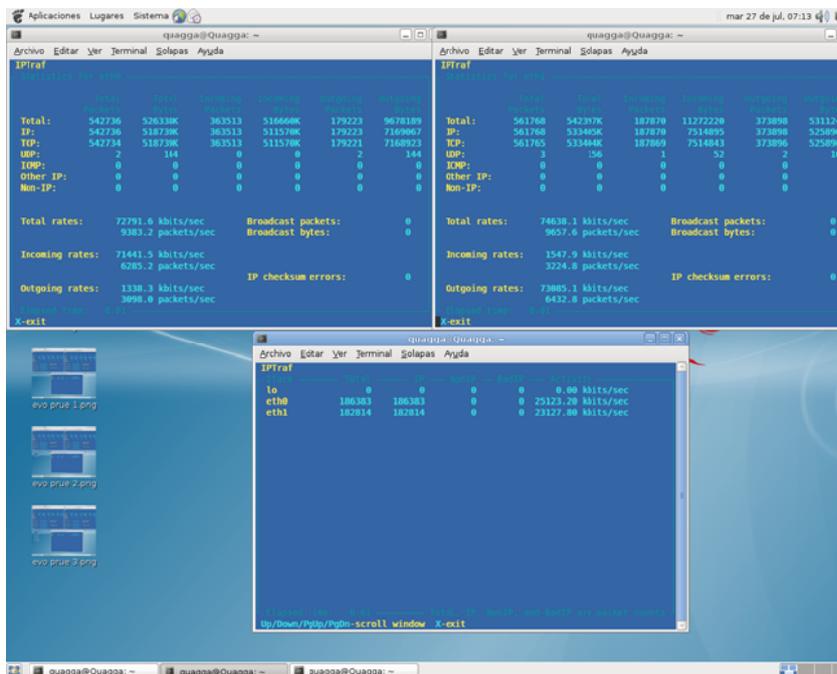


Ilustración 33: Prueba 7 IPTraf, EVO.

Octava Prueba

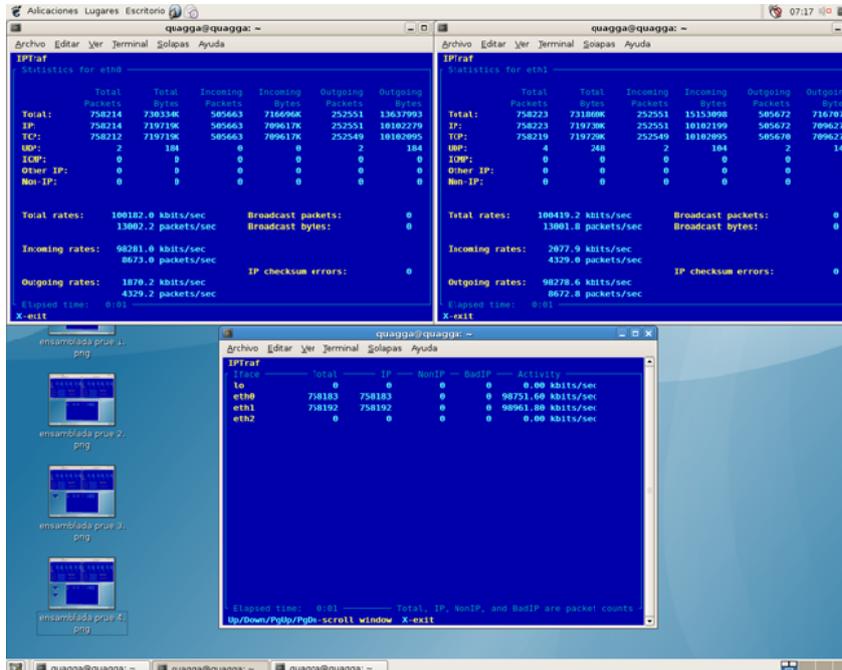


Ilustración 34: Prueba 8 IPTraf, Ensamblada.

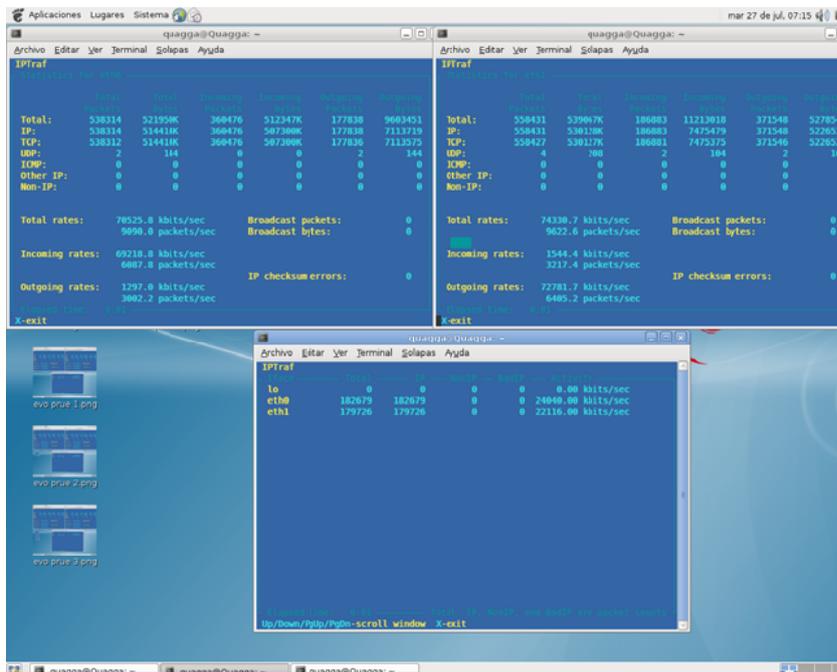


Ilustración 35: Prueba 8 IPTraf, EVO.

Si comparamos cada una de las pruebas veremos que solo en la primera la EVO le ganó a la Ensamblada, pero en las 7 restantes la ensamblada superóampliamente a la EVO ya sea en Kbits/seg o en packets/seg, demostrando un mejor desempeño.

### **Análisis de transferencia**

En esta última fase se transmitió un archivo a través de los 3 equipos, este archivo fue una imagen de un DVD en formato ISO, con un tamaño de 1.46 GB (1,533,024 KB), para determinar el tiempo en que se transmite el archivo, quedando de la siguiente manera.

**Tabla 6: Transferencia ISO.**

Equipo	Tipo de archivo	Tiempo
EVO	Imagen ISO	2.33 Min
Ensamblada	Imagen ISO	2.35 Min
Ruteador 2800	Imagen ISO	2.34 Min

Viendo los resultados tan parecidos, se optó por hacer otra transferencia de archivos pero ahora con 2 máquinas transmitiendo al mismo tiempo hacia una máquina, y así crear más tráfico en la red.

Se procedió a extraer los archivos de la imagen ISO, una vez que se extrajeron los archivos se pasaron a la 2 máquinas para empezar con la segunda transmisión de archivos, obteniendo los siguientes resultados.

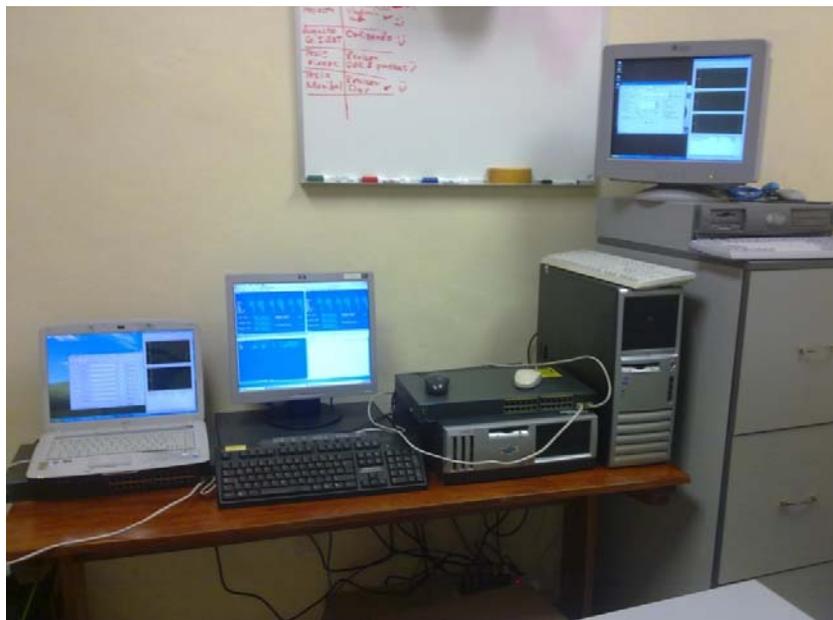
**Tabla 7: Transferencia ISO y Archivos.**

Equipo	Tipo de archivo	Tiempo
Ruteador 2800	Archivos	4.15 Min
	Imagen ISO	4.28 Min
Ensamblada	Archivos	4.16 Min
	Imagen ISO	4.28 Min
EVO	Archivos	4.22 Min
	Imagen ISO	4.29 Min

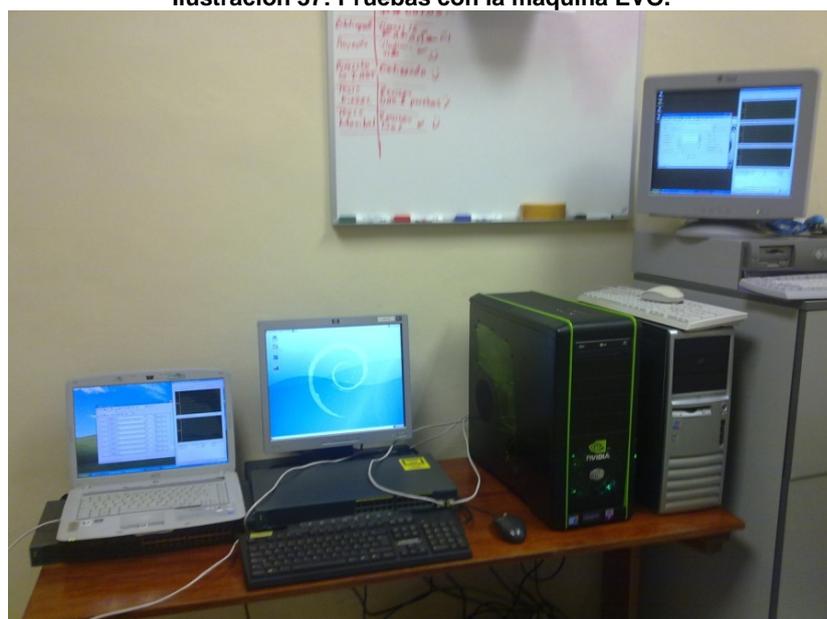
Con esto se termina el proyecto, desde el inicio, desarrollo y los resultados del mismo.



**Ilustración 36: Pruebas con el ruteador 2811.**



**Ilustración 37: Pruebas con la máquina EVO.**



**Ilustración 38: Pruebas con la máquina ensamblada.**

# Capítulo 5

---

## Conclusiones

En esta sección del trabajo se explicará brevemente en que consistió el proyecto junto con los resultados.

Para iniciar, se investigó la estructura de la red informática de la Universidad de Quintana Roo, en qué forma esta interconectada a su proveedor, en este caso Telmex, cuantos enlaces tiene, que protocolo usan, los equipos que usan para los enlaces, tipos de conectores entre los equipos y lo necesario hacer funcionar la RIU.

De igual forma se investigó en el Internet qué soluciones de enrutamiento existen y que sean de distribución gratuita, cuáles son sus ventajas y desventajas de los mismos, de igual forma se analizó los sistemas operativos de GNU/Linux para su implementación del programa de enrutamiento, quedando al final el programa de Quagga y el sistema operativo Debian.

Una vez que se determinó el programa de enrutamiento a utilizar, se consiguieron los equipos que se utilizarían para el desarrollo del proyecto, en un principio sólo se contaba con la máquina ensamblada y un ruteador Cisco 2811, por lo cual se habló con el encargado la carrera y se consiguieron 2 Switches Cisco modelo 2960, 2 ruteadores Ciscos modelos 2811, una máquina COMPAQ modelo EVO, así como una máquina HP modelo DC7600; los equipos anteriores fueron prestados por la carrera de Ingeniería en Redes, también se consiguió una laptop Acer modelo Aspire 5520, y una máquina ensamblada, la cual fue adquirida para el proyecto con un alto desempeño. A continuación se procedió a la instalación de los mismos,

Dado que se tenían los equipos, se procedió a la instalación del sistema operativo *Debian* a 2 de las máquinas, a la EVO y a la ensamblada, teniendo el OS instalado en ambos equipos, se procedió a instalar y configurar el Quagga; cuando se terminó con la configuración se continuó con los 2 ruteadores, dado que también requieren configuración para poder trabajar correctamente.

Teniendo todo configurado, se procedió a interconectar los equipos entre sí para empezar con las pruebas de conectividad, para ver que el Quagga esté funcionando correctamente y que puede enrutar los paquetes. Posteriormente, se aplicó una política de Netfilter con iptables en el Debian, para que funcione el NAT en conjunto al Quagga, dándole acceso a una máquina a Internet. Debido a que la configuración que se instaló en los equipos no corresponde a la de la institución, denegando el acceso a Internet por políticas de seguridad, dando como opción el NAT.

En esta segunda parte de las pruebas, se consiguió un programa de inyección de paquetes, para poder medir el desempeño de los 3 equipos de Ruteo, verificando cuantos paquetes por segundo soportaba cada uno de ellos.

La recolección de los datos se haría de 2 formas, la primera y principal mediante el programa D-ITG el cual nos servirá como inyector de paquetes, pero también nos permitirá sacar estadísticas de los resultados del envío de paquetes, mediante un módulo que tiene integrado el cual le permite sacar los registros, en los cuales almacena la información de la transferencia que se realizó, este programa se instaló en los 2 equipos restantes, en la máquina Acer y en la HP, los dos con el sistema operativo Windows XP Profesional, en la cual la Acer será la que envíe los paquetes y la HP la que los reciba, configurando la HP para generar los archivos registros.

La segunda parte será mediante un programa llamado IPTrafel cual será instalado en las máquina con GNU/Linux, como quien dice en la ensamblada y en la EVO, dicho programa nos permitirá ver en tiempo real el desempeño de

las interfaces de red, por ejemplo, a qué velocidad de transferencia está trabajando, así como, cuantos paquetes por segundo se están transfiriendo.

En cuanto a la inyección de paquetes se realizó en 8 pruebas por igual a los 3 equipos, tomando 60 segundos de envío en todas las pruebas, de la cual la primera sería con 1000 paquetes por segundo, con un tamaño de 512 bytes por paquete, haciendo de cada paquete de 0.5 KB, sin embargo la quinta prueba sería lo contrario, mandando a 9,999,999 paquetes por segundo con un tamaño de 8192 bytes por paquete, haciendo de cada paquete de 8 KB, las pruebas 6, 7 y 8, tendrán la mismas especificaciones de la quinta, pero con diferentes envíos a la vez; la sexta sería a 2 envíos a la vez, la séptima a 4 envíos simultáneos y la octava sería a 8 envíos al mismo tiempo.

Ya que se terminaron las 8 pruebas a los 3 equipos, se procedió a extraer los archivos de registro, los cuales fueron guardados en la máquina con un nombre referente al equipo y a la prueba que se realizó, ejemplo: *ensamblada pru 1*, con el fin de analizar los resultados y sacar gráficas con ellas, haciendo más fácil verificar el desempeño de cada máquina según sea la prueba.

También, como se mencionó anteriormente se había instalado el IPTraf en las máquinas, que tienen instalado el Quagga, las cuales se extrajeron imágenes de las pantallas cuando se realizó la inyección de paquetes, al segundo 59 de la prueba, teniendo 3 ventanas en cada imagen, dos de las cuales mostraban el desempeño de cada interfaz, y la tercera con el concentrado de las 2 interfaces.

Con esto se podría ver la comparativa entre las dos máquinas, desde el flujo de paquetes hasta la velocidad de transferencia en las 2 interfaces, de entrada y salida de los paquetes.

Teniendo los resultados finales, se muestra que los 3 equipos pueden soportar casi la misma carga de trabajo, es poca la diferencia que hay entre los resultados entre las 3 máquinas, en la primera los 3 equipos pasaron la misma

cantidad de información, en la segunda prueba mandó más información el ruteador, pero en la 3, 4 y 5 prueba logró mandar más información la máquina EVO, en el resto de las pruebas la ensamblada sobresalió ligeramente de los otros 2 equipos.

Por lo tanto mediante los resultados obtenidos en las pruebas hasta este punto, podemos ver que si funcionaría el Quagga como un sistema de respaldo o sustitución del ruteador.

A mi parecer, los resultados parecidos se pudieron deber a un cuello de botella que se hubiera creado en alguna sección de la red, teniendo en cuenta que el ruteador es un equipo dedicado para ese flujo de información, la máquina ensamblada es de más actual en el mercado, y por último la EVO siendo un equipo antiguo con menos recursos pudo compararse con los otros 2.

Para las pruebas que se recolectaron con el IPTraf, podemos ver que la máquina ensamblada sobresale por casi por una cuarta parte a la EVO. Demostrando que la ensamblada puede transmitir más paquetes por segundo que la EVO, dado que no se podría comparar una máquina que fue adquirida en el año 2000, a una que se adquirió hace 7 meses.

En la última fase de las pruebas, la transmisión de información por volumen, en la primera ronda con la imagen ISO, los 3 equipos pasaron casi al mismo tiempo el archivo, pero en la segunda ronda en él envió de los 2 archivos, el ruteador ganó por un poco, en segundo lugar la ensamblada, pero la EVO quedó un poco atrasado en comparación a los otros 2 equipos.

Mostrado todo lo anterior, una máquina puede hacer el mismo trabajo que un ruteador sin gastar tanto dinero, igual de fácil que administrar un ruteador Cisco, y sobre todo más fácil la adquisición y conseguir las refacciones de equipo, tan solo en ir a la tienda de informática de que uno guste.

## Bibliografía

1. **McDonald., Wendell Odom y Rick.***Routers and routing basics CCNA 2 companion guide.* Indianapolis : Pearson Education. , 2007.
2. **Sportack., Mark A.***Fundamentos de enrutamiento IP.* Madrid : Pearson Educacion; Cisco Press., 2003. pág. 354 p.
3. **Schlegel., Anna Navarro.***Diccionario de términos de comunicaciones y redes.* Madrid : Cisco Press; Pearson Educacion, 2003. . pág. 612 p.
4. **Hill., Brian.***Cisco : manual de referencia.* Madrid : McGraw Hill, 2002. .
5. **Lorenz., Jim.***Academia de networking de Cisco Systems: prácticas de laboratorio.* Madrid : Pearson Educación; Cisco Press, 2002. .
6. **Cisco Systems, Inc., Cisco Networking Academy Program.***Cisco Networking Academy Program. second-year companion guide.* Indianapolis, IN : Cisco Press., 2001. pág. 719 p.
7. **Pinsky., Allan Leinwand y Bruce.***Configuración de Routers Cisco.* Madrid : Cisco Press., 2001. pág. 356 p.
8. **McPherson., Sam Halabi y Danny.***Arquitecturas de enrutamiento en Internet.* . Madrid : Pearson Educación; Cisco Press. , 2001. .
9. **Habraken., Joe.***Routers Cisco.* Madrid. : Prentice Hall., 2000. pág. 420 p.
10. **Chappell., Editado por Laura.***Advanced Cisco router configuration.* Indianapolis. : Cisco Systems., 1999. pág. 644 p.
11. **Sportack., Mark A.***IP routing fundamentals.* Indianapolis. : Cisco Systems,Cisco Press., 1999. pág. 510 p.
12. Escuela de Computación de la Universidad Central de Venezuela. *Escuela de Computación de la Universidad Central de Venezuela.* [En línea] Noviembre

de 2009. [Citado el: 12 de Julio de 2010.]

<http://www.ciens.ucv.ve/escueladecomputacion/documentos/archivo/86>.

13. **Inc, IP Infusion.** GNU Zebra. *GNU Zebra* . [En línea] IP Infusion Inc, 2003. [Citado el: 20 de Febrero de 2010.] <http://www.zebra.org/>.

14. **OpenBGPD.** OpenBGPD 4.6. *OpenBGPD 4.6*. [En línea] OpenBGPD , 2004-2009. [Citado el: 20 de febrero de 2010.] <http://www.openbgpd.org/>.

15. XORP . *The XORP Team*. [En línea] XORP , Enero de 2009. [Citado el: 20 de febrero de 2010.] <http://www.xorp.org/>.

16. **Filip, Ondrej.** The BIRD Internet Routing Daemon. *The BIRD Internet Routing Daemon*. [En línea] The BIRD Internet Routing Daemon, 03 de Agosto de 2010 . [Citado el: 20 de Febrero de 2010.] <http://bird.network.cz/>.

17. Quagga Routing Suite. *Quagga Routing Suite*. [En línea] Quagga, 19 de Agosto de 2010. [Citado el: 20 de Febrero de 2010.] <http://www.quagga.net/>.

18. Debian. *Debian*. [En línea] Debian, 9 de Septiembre de 2010. [Citado el: 24 de Febrero de 2010.] <http://www.debian.org/index.es.html>.

19. **Izura, Pello Xabier Altadill.** IPTABLES Manual práctico. *IPTABLES Manual práctico*. [En línea] [Citado el: 05 de Junuo de 2010.] <http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall.pdf>.

20. **Ayuso, Pablo Neira.** netfilter. *netfilter*. [En línea] netfilter, 2010. [Citado el: 05 de junio de 2010.] <http://www.netfilter.org/>.

21. **Il, 'Federico.** D-ITG, Distributed Internet Traffic Generator . *D-ITG, Distributed Internet Traffic Generator* . [En línea] D-ITG, Mayo de 2007. [Citado el: 25 de Julio de 2010.] <http://www.grid.unina.it/software/ITG/>.

22. Graphical User Interface for D-ITG 2.6. *Graphical User Interface for D-ITG 2.6*. [En línea] [Citado el: 25 de Julio de 2010.] <http://www.semken.com/projekte/index.html>.

## **Glosario**

Prompt: Indicador de donde está ubicado en el router u OS en la línea de comando; ejemplo: modo usuario \$, modo súper usuario #.

Vty: Se trata de un conjunto de puertos virtuales utilizados para la conexión vía telnet, SSH, http o https al dispositivo para realizar administración in band. Mayormente usado en los routers.

Router: También conocido como router, es un dispositivo que permite interconectar las redes, uniéndose entre ellos para hacer una red más grande. Como si fuera una telaraña inmensa.

E1: Es un servicio de red para conectarse hacia Internet, siendo un enlace dedicado y teniendo un máximo de ancho de banda de 2.048 Mbps, La Jerarquía Digital Plesiócrona, conocida como PDH.

E3: Es un servicio de red para conectarse hacia Internet, siendo un enlace dedicado y teniendo un máximo de ancho de banda de 34.368 Mbps, su jerarquía es igual a la de la E1.

MPLS: Es una red privada que combina la flexibilidad de las comunicaciones punto a punto siendo fiable en su calidad y seguridad de los servicios Private Line, FrameRelay o ATM, al igual ofrece niveles de rendimiento diferenciados y priorización del tráfico, así como aplicaciones de voz y multimedia. Y todo ello en una única red.

**Switch:**Dispositivo usado para interconectar dos o más segmentos de red, de manera similar a los puentes (bridges), pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red.

**Conmutación:** Es el trabajo que realizan los ruteadores, el envío de paquetes de un lugar a otro.

**GNU/Linux:** Es un sistema operativo de distribución gratuita, el cual tiene muchas variantes, está basado en el Unix lo cual le brinda cierta seguridad y fiabilidad.

**Ruteador:** Es el nombre el Router en Español.

**Snifers ó Sniffers:** Software que puede monitorear una red, o capturar paquetes con el fin de analizar qué tipo de tráfico está pasando por dicha red.

**Protocolo:**Es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales. En su forma más simple, un protocolo puede ser definido como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación.

**Firewall:** Es un software o hardware utilizado en redes de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas.

**Memoria RAM:** Es donde la máquina guarda los datos que se está utilizando en ese momento,el almacenamiento se consideratemporal dado que los datos y programas permanecen en ella mientras que la máquina este encendida o no sea reiniciada.

**Disco Duro:**Es donde se guarda la información del usuario al igual que el OS y no se borra con la pérdida de energía, emplea un sistema de grabación magnética digital.

**Tarjetas de Red:**El dispositivo más utilizado en estos momentos para conectar un dispositivo a red,ya sea por conexión física o inalámbrica.

ACL: Es una tabla que le dice a un sistema los derechos de acceso que posee cada usuario para un objeto determinado, como directorios, archivos, puertos, etc.

Puerto: Es una forma de denominar a una interfaz a través de la cual los diferentes tipos de datos se pueden enviar y recibir. Dicha interfaz puede ser de tipo físico, o puede ser a nivel de software (por ejemplo, los puertos que permiten la transmisión de datos entre diferentes ordenadores, como ftp, http, etc.)

Ancho de banda: Es la cantidad de información o de datos que se puede enviar a través de una conexión de red en un período de tiempo dado, se indica generalmente en bits por segundo (bps), kilobits por segundo (Kbps), o megabits por segundo (Mbps).

Quagga: Es un software libre para emular a un ruteador, está diseñado especialmente para NetBSD, FreeBSD, Solaris y GNU/Linux.

Root: Es el nombre convencional de la cuenta de usuario que posee todos los derechos, root es también llamado súper usuario, normalmente ésta es la cuenta de administrador.

Demonio: Es un tipo de proceso especial que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).

Comando: Es una instrucción u orden que el usuario proporciona a un sistema informático, desde la línea de comandos (como una Shell o MS-DOS) o desde una llamada de programación. Puede ser interno (contenido en el propio intérprete) o externo (contenido en un archivo ejecutable).

Dirección IP: Etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de red (elemento de comunicación/conexión) de un dispositivo

(habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del protocolo TCP/IP.

Local host: Es un nombre reservado que tienen todas las computadoras, router o dispositivo independientemente de que disponga o no de una tarjeta de red Ethernet. El nombre local host es traducido como la dirección IP de loopback 127.0.0.1 en IPv4, o como ::1 en IPv6.

Loopback: El dispositivo de red loopback es un interfaz de red virtual que siempre representa al propio dispositivo independientemente de la dirección IP que se le haya asignado. El valor en IPv4 es 127.0.0.1 y ::1 para el caso de IPv6.

NAT: (Network Address Translation - Traducción de Dirección de Red) es un mecanismo utilizado por Routers IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles. Consiste en convertir en tiempo real las direcciones utilizadas en los paquetes transportados.

iptables: herramienta de cortafuegos en GNU/Linux, que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT) para IPv4 y mantener los registros.

Interfaz: es el puerto (circuito físico) a través del que se envían o reciben señales desde un sistema o subsistemas hacia otros. No existe una interfaz universal, sino que existen diferentes estándares (Interfaz USB, interfaz SCSI, etc.) que establecen especificaciones técnicas concretas (características comunes), con lo que la interconexión sólo es posible utilizando la misma interfaz en origen y destino.

Segmento de red: Es el conjunto de máquinas de una red, definidos por un bloque de direcciones IPs, según sea la máscara de la configuración de la red.

**Boot:** La secuencia de arranque, es el proceso que inicia el sistema operativo cuando el usuario enciende una computadora. Se encarga de la inicialización del sistema y de los dispositivos.

**DHCP:**Dynamic Host Configuration Protocol - Protocolo de configuración dinámica de host, se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP, generalmente dinámicas, y las va asignando a los clientes, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

**Gateway:** Puerta de enlace, es normalmente un equipo configurado para dotar a las máquinas de una red local (LAN) conectadas a él de un acceso hacia una red exterior, generalmente realizando para ello operaciones de traducción de direcciones IP "NAT".

**DNS:**Base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

**GRUB:** es un administrador o gestor de arranque múltiple, GRUB en español: Gran Gestor de Arranque Unificado, que se usa comúnmente para iniciar uno de dos o más sistemas operativos instalados en una misma computadora.

**PCI:** "Interconexión de Componentes Periféricos" consiste en un bus de computadora estándar para conectar dispositivos periféricos directamente a su placa base.

**GHz:** El gigahercio (GHz) es un múltiplo de la unidad de medida de frecuencia hercio (Hz) y equivale a 10<sup>9</sup> (1.000.000.000) Hz

DDR3: es un tipo de memoria RAM. Forma parte de la familia SDRAM de tecnologías de memoria de acceso aleatorio, que es una de las muchas implementaciones de la SDRAM.

MS-DOS: es un sistema operativo perteneciente a la familia DOS comercializado por Microsoft para el IBM PC.

TXT: Los archivos de texto plano, son aquellos que están compuestos únicamente por texto sin formato, sólo caracteres. Estos caracteres se pueden codificar de distintos modos dependiendo de la lengua usada.

Registro: Es un registro de actividad de un sistema, que generalmente se guarda en un archivo de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema.

Imagen ISO: Archivo donde se almacena una copia o imagen exacta de un sistema de archivos, normalmente un disco compacto, un disco óptico, como un CD, un DVD.

DVD: disco de video digital, es un dispositivo de almacenamiento óptico cuyo estándar surgió en 1995.

Plesiócrona: significa *casi síncrono* y ha sido definido por la ITU de la forma siguiente: "Dos señales son plesiócronas si sus instantes significativos correspondientes se presentan con la misma cadencia nominal, y cualquier variación de esta cadencia se mantiene dentro de unos límites especificados. De esta forma, dos señales que tengan la misma velocidad digital nominal y que no provengan del mismo reloj, serán generalmente plesiócronas".

## **Abreviaturas**

Universidad de Quintana Roo (UQROO).

Multi Protocolo mediante Etiquetas (MPLS).

Red Privada Virtual (VPN).

Teléfonos de México (TELMEX).

Tecnologías de la Información y la Comunicación (TIC).

Centro Universitario de Computo (CUC).

Sistema Operativo (OS).

Red Informática Universitaria (RIU).

Redes de Área Amplia (WAN).

Modo de Transferencia Asíncrona(ATM).

Línea de Abonado Digital Asimétrica (ADSL).

Línea Digital de Abonado de alta velocidad (HDSL).

Red Digital de Servicios Integrados (RDSI).

Lista de Control de Acceso (ACL).

Traducción de Dirección de Red(NAT).

Línea de Suscriptores Digitales (DSL).

Memoria de Acceso Aleatorio (RAM)

Tarjeta de Interfaz de WAN(WIC).

Protocolo de Encaminamiento de Información (RIP)

Protocolo de Enrutamiento de Gateway Interior (IGRP).

Protocolo Extendido de Enrutamiento de Gateway Interior (EIGRP).

Abrir más corta primero la ruta de acceso (OSPF).

Protocolo de Encaminamiento Externo (BGP).

Megabyte (MB).

Megabit (Mb).

Registro Federal de Contribuyentes (RFC).

Sin Número (S/N).

Colonia (Col).

Disco Compacto (CD).

Protocolo de Internet (IP).

Protocolo Punto a Punto (PPP).

Ethernet (eth).

Red de Área Local (LAN).

Sistema Operativo de Interconexión de Redes (IOS).

Protocolo de Control de Transmisión (TCP).

Sistema de Nombre de Dominio (DNS).

Protocolo de Configuración Dinámica de Host (DHCP).

Controlador Disco Duro (HDD).

Gran Gestor de Arranque Unificado (GRUB).

Interconexión de Sistemas Abiertos (OSI).

Unión Internacional de Telecomunicaciones (ITU).

Organización Internacional para la Estandarización (ISO).

Par Trenzado no Blindado (UTP).

Chequeo de Redundancia Cíclica (CRC).

Control de Acceso al Medio (MAC).

Protocolo de Datagrama de Usuario (UDP).

Intercambio de Paquetes Secuenciados (SPX).

Llamada a Procedimiento Remoto (RPC).

Protocolo de Transferencia de Archivos (FTP).

Protocolo Simple de Transferencia de Correo (SMTP).

Protocolo de Transferencia de Híper Texto (HTTP).

Protocolo de Transferencia de Híper Texto Seguro (HTTPS).

# **Anexo A**

## Instalación de Quagga

En ocasiones pide el paquete para poder instalar el Quagga, para instalar estos paquetes se usa el comando ***apt-get install***, y mediante el CD de instalación del OS que se esté manejando, en ocasiones no se tiene el CD del OS, para esos momentos tenemos la Internet, con tan solo configurar una tarjeta de red para tener acceso a Internet.

Nota: se recomienda realizar las instalaciones y configuraciones con el usuario ***root***, para evitar problemas o errores con los permisos, para entrar en este modo depende del OS, por lo general con el comando "***su -***"o ***sudo*** este se utiliza en Ubuntu, a continuación pide un *password*, el cual definimos en la instalación del OS.

Si pide una serie de complementos de ***g++*** o ***gcc***, para poder instalar el Quagga, se pone el comando:

```
apt-get install libc6-dev g++ gcc
```

Otro complemento que requiere el Quagga para poder ser instalado es el *libcap1*, el cual se puede instalar con el siguiente comando:

```
apt-get install libcap1
```

Con esto se instalará una serie de complementos requeridos para el Quagga.

Cuando no tenemos el CD de instalación, pero si una red que nos da acceso a Internet, tenemos la opción de modificar un registro, para decirle al OS debían, que en lugar de buscar los paquetes en el CD, los busque en una página de Internet previamente configurada.

Para lograr esto modificamos el archivo:

```
vim /etc/apt/sources.list
```

En el podemos agregar sitios de donde podemos obtener los paquetes, por default están agregados unos, pero deshabilitado, para habilitarlos les quitamos el # con el cual inicia la línea.

A continuación se mostrara un ejemplo del archivo sources.list:

```
#
# debcdrom:[Debian GNU/Linux 5.0.4 _Lenny_ - Official i386 CD Binary-1
20100131
-18:53]/ lenny main

#deb cdrom:[Debian GNU/Linux 5.0.4 _Lenny_ - Official i386 CD Binary-1
20100131-
18:53]/ lenny main

deb http://security.debian.org/ lenny/updates main
deb-src http://security.debian.org/ lenny/updates main

deb http://volatile.debian.org/debian-volatile lenny/volatile main
deb-src http://volatile.debian.org/debian-volatile lenny/volatile main
```

Después de modificar el registro salimos con la tecla **Esc** y poniendo **:wq** con el cual indicamos que guarde y salga del registro.

Ponemos `apt-getupdate` para que el OS se actualice el sistema de acceso a las programas.

En ocasiones se podrá instalar la aplicación **rcconf**, para poder cargar las aplicaciones como demonio y que inicien junto al debían. Para lograr eso ponemos las siguientes líneas.

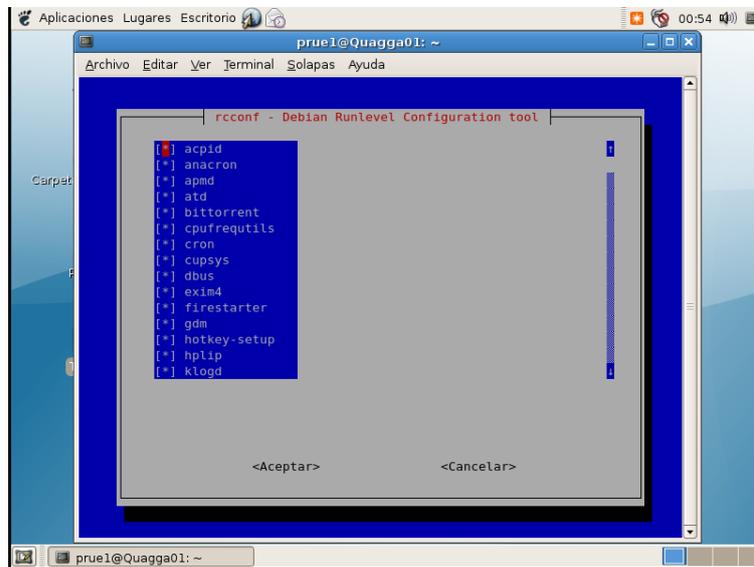
```
apt-getinstallrcconf
```

Ya que este instalado el **rcconf** se podrá verificar que el servicio de Quagga quede como demonio, así como otros demonios que se encuentren el servidor.

```
Quagga:~#rcconf
```



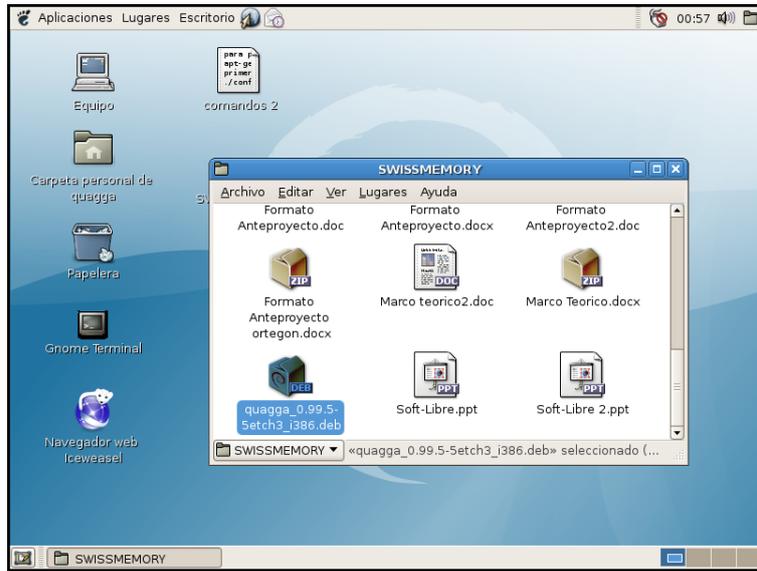
**Correr el programa rcconf.**



#### Programas que inician con el rconf.

Para adquirir el programa de Quagga entramos a la página de [www.quagga.net](http://www.quagga.net), seleccionamos el paquete para debían y para la plataforma i386, a continuación lo descargamos; ya que se descargó ponemos el comando **dpkg -imás el nombre del archivo**, para poder descomprimir e instalar el paquete de Debian.

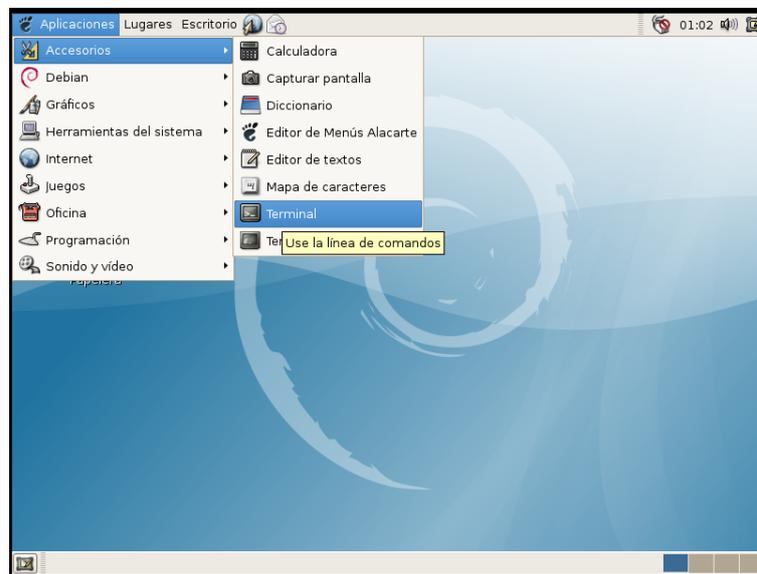
Algunas cosas se pueden hacer mediante comando y otras de manera grafica, mover archivos o carpetas, renombrar archivos o por el estilo; algunas otras tendrán que ser por comandos, como configurar el Quagga.



### Mover el programa a la máquina.

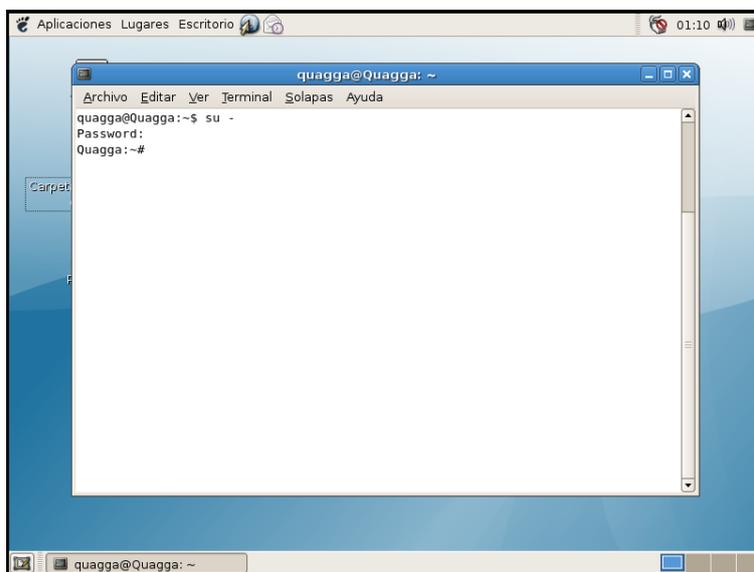
Primero seleccionamos el programa de Quagga y lo copiamos al disco duro, con el fin de tener acceso al programa más fácilmente.

Una vez teniendo el archivo en el disco duro abrimos una ventana de *Terminal*.



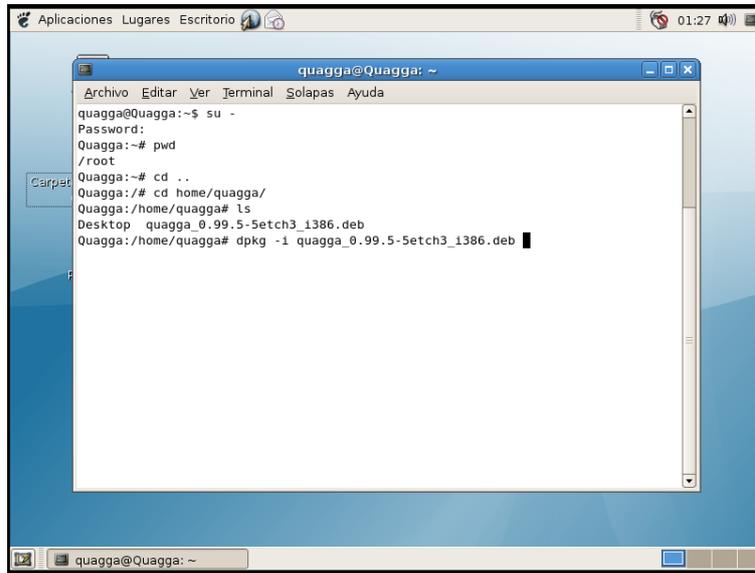
### Cómo entrar a la termina.

Una vez abierta la terminal, ponemos `su -`, con el fin de acceder como root para no tener problemas con la instalación el Quagga, pedirá un *password* el cual es el de súper usuario.



**Acceder como súper usuario.**

A continuación instalaremos el programa, estando en la terminal nos dirigimos a la ubicación del archivo que copiamos a la máquina, estando en la ubicación tenemos que poner ***dpkg -i***, agregando el nombre completo del archivo, con el cual instalará Quagga.

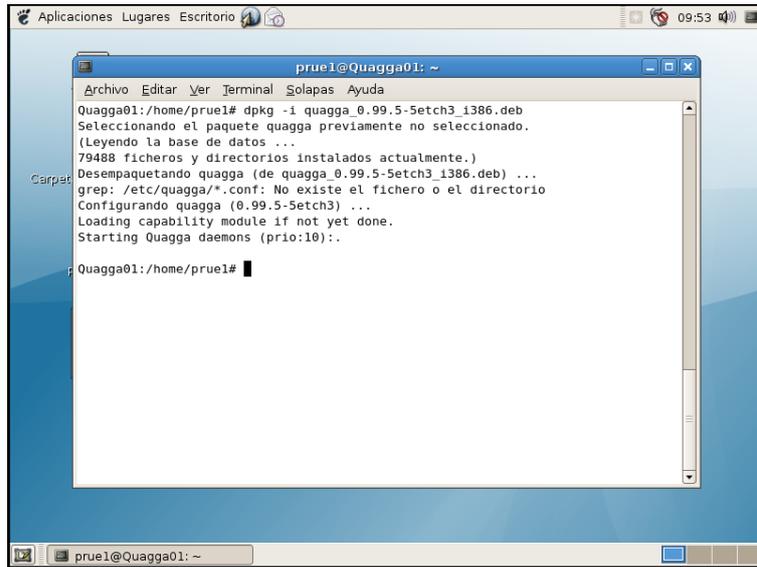


```
quagga@Quagga: ~  
Archivo Editar Ver Terminal Solapas Ayuda  
quagga@Quagga:~$ su -  
Password:  
Quagga:~# pwd  
/root  
Quagga:~# cd ..  
Quagga:~/# cd home/quagga/  
Quagga:/home/quagga# ls  
Desktop quagga_0.99.5-5etch3_1386.deb  
Quagga:/home/quagga# dpkg -i quagga_0.99.5-5etch3_1386.deb
```

### Instalar Quagga.

```
dpkg -i quagga_0.99.5-5etch3_1386.deb
```

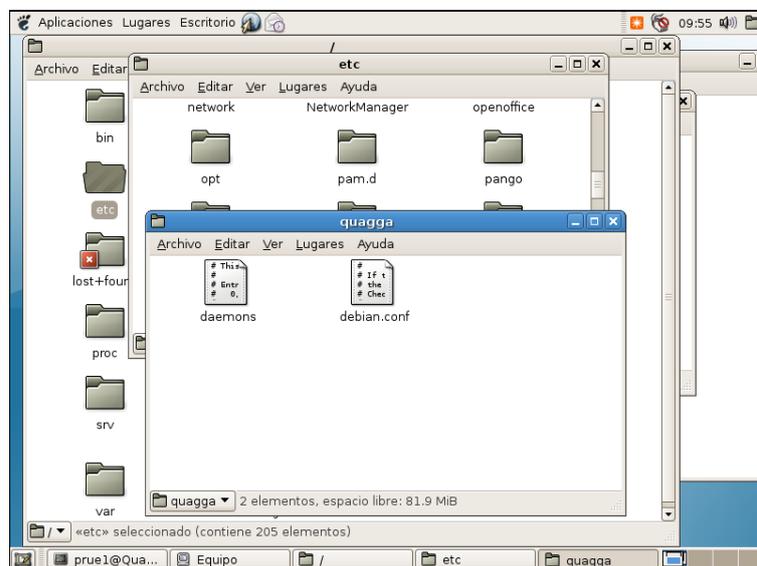
Al terminar mostrará una leyenda diciendo que ya terminó satisfactoriamente, otra forma es teniendo acceso a Internet, ponemos el siguiente comando *apt-get install quagga*, con el cual buscará en el CD o en Internet el programa para que se instale.



**Cómo muestra el fin de la instalación.**

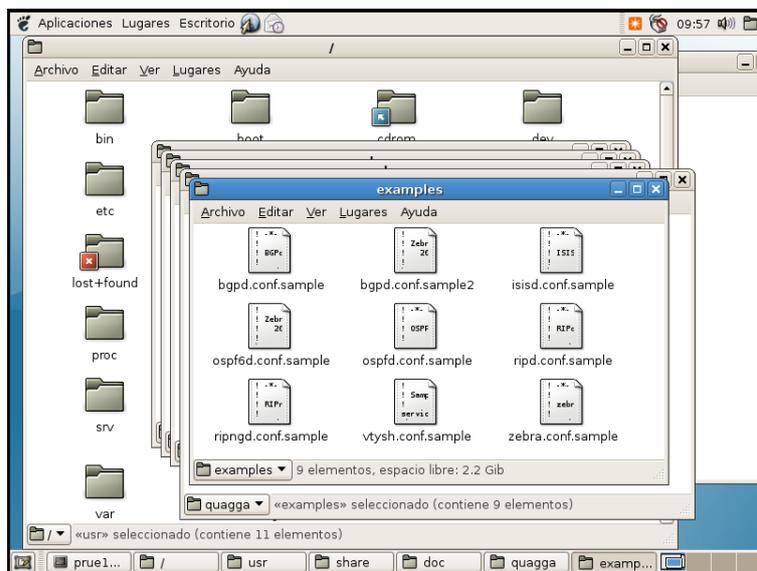
Una vez que se terminó con la instalación de Quagga verificamos que exista la carpeta, en la cual solo contendrá 2 archivos:

/etc/quagga/



**Carpeta de Quagga después de su instalación.**

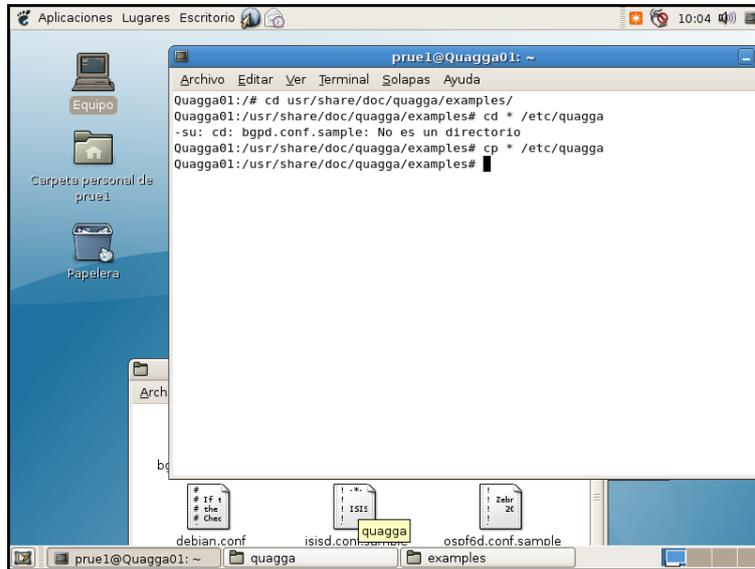
Ya que esta la carpeta nos dirigimos a esta ruta `/usr/share/doc/quagga/examples/` donde encontraremos varios archivos los cuales copiaremos todos a la carpeta antes creada `/etc/quagga/`.



**Carpeta de ejemplos de Quagga.**

```
cd /usr/share/doc/quagga/examples/
```

```
cp * /etc/quagga/
```



**Cómo copiar los archivos a la carpeta de Quagga.**

# **Anexo B**

## Daemons

```
# This file tells the quagga package which daemons to start.
#
# Entries are in the format: <daemon>=(yes|no|priority)
# 0, "no" = disabled
# 1, "yes" = highest priority
# 2 ..10 = lower priorities
# Read /usr/share/doc/quagga/README.Debian for details.
#
# Sample configurations for these daemons can be found in
# /usr/share/doc/quagga/examples/.
#
# ATTENTION:
#
# When activation a daemon at the first time, a config file, even if it
is
# empty, has to be present *and* be owned by the user and group
"quagga", else
# the daemon will not be started by /etc/init.d/quagga. The permissions
should
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It should be
owned by
# group "quaggavty" and set to ug=rw,o= though. Check
/etc/pam.d/quagga, too.
#
zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=no
isisd=no
```

## Debian.conf

```
#
# If this option is set the /etc/init.d/quagga script automatically
loads
# theconfig via "vtysh -b" when the servers are started.
# Check /etc/pam.d/quagga if you intend to use "vtysh"!
#
vtysh_enable=yes
zebra_options=" --daemon -A 127.0.0.1 192.168.8.157 10.10.100.254
10.10.200.254"
bgpd_options=" --daemon -A 127.0.0.1"
ospfd_options=" --daemon -A 127.0.0.1"
ospf6d_options="--daemon -A ::1"
ripd_options=" --daemon -A 127.0.0.1"
ripngd_options="--daemon -A ::1"
isisd_options=" --daemon -A 127.0.0.1"
```

## Ripd.conf

```
!  
! Zebra configuration saved from vty  
!   2010/07/20 11:36:43  
!  
hostname quagga_rip  
password quagga  
enable password quagga  
logstdout  
!  
router rip  
version 2  
network 10.10.100.0/24  
network 10.10.200.0/24  
network 192.168.8.0/24  
network eth0  
network eth1  
network eth2  
!  
line vty  
!
```

## Zebra.conf

```
!  
! Zebra configuration saved from vty  
!   2010/07/20 11:32:25  
!  
hostname Quagga  
password quagga  
enable password quagga  
!  
interface eth0  
description WAN  
ip address 192.168.8.157/24  
ipv6nd suppress-ra  
!  
interface eth1  
description 100.254  
ip address 10.10.100.254/24  
ipv6nd suppress-ra  
!  
interface eth2  
description 200.254  
ip address 10.10.200.254/24  
ipv6nd suppress-ra  
!  
interface lo  
!  
interface sit0  
ipv6nd suppress-ra  
!  
ip route 0.0.0.0/0 192.168.8.254  
!  
router-id 192.168.8.157
```

!  
linevty  
!