PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN DE GRADO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

TEMA: "Diseño e Implementación de Software Gráfico para la Administración y Control de Ancho de Banda y Seguridades en Redes basado en Linux."

AUTORA:

PAOLA CRISTINA PADILLA ALBAN

QUITO, NOVIEMBRE 2010

DEDICATORIA

Este trabajo va dedicado a mis padres, que con su esfuerzo y sacrificio hicieron posible la culminación de mis estudios y también a mi esposo que con su apoyo y comprensión me ha ayudado a alcanzar mis metas.

AGRADECIMIENTO

Agradezco a Dios por todas sus bendiciones, a toda mi familia por su apoyo, especialmente a mi tío Diego, que fue mi mejor amigo durante mi vida universitaria el cual me alentó y guio para salir siempre adelante. A mí cuñada Anita por toda su ayuda y apoyo incondicional.

Un especial agradecimiento a mi director Ing. Alfredo Calderón y a mis correctores Ing. Xavier Cóndor e Ing. Francisco Rodríguez que con su experiencia y sabiduría hicieron posible la culminación de este trabajo.

CONTENIDO

LISTA DE GRÁFICOS

LISTA DE CUADROS

RESUMEN

El presente proyecto de titulación se enfoca en la creación de un software gráfico para el control de ancho de banda de una red. El mismo fue desarrollado e implementado en un servidor con sistema operativo GNU/Linux.

La interfaz gráfica fue desarrollada para ser utilizada mediante la web, por lo que está diseñada en html y php. Además está basada en herramientas de GNU/Linux que complementan al aplicativo como son: CBQ, IPtables, Squid, MRTG y DansGuardian.

El aplicativo puede ser utilizado en cualquier empresa mediana o pequeña en la cual se requiera utilizar de forma productiva el ancho de banda.

Este trabajo incluye una guía completa de instalación de cada una de las herramientas que fueron utilizadas para generar el software.

INTRODUCCIÓN

En los últimos tiempos el Internet se ha convertido en uno de los principales medios de comunicación.

Cada vez son más las empresas e instituciones que lo utilizan por lo que los usuarios se incrementan con una rapidez asombrosa. Por este crecimiento vertiginoso es necesario que se tomen medidas que permitan el control y correcto mantenimiento de las redes de datos, ya sea realizar subredes, dividir direcciones en públicas y privadas o utilizar direccionamiento con Ipv6.

Estas soluciones, permiten que el crecimiento del número de usuarios no se vea afectado por varios años, sin embargo, por la cantidad de usuarios y aplicaciones en la Web que se pueden presentar, se debe tomar otro tipo de medidas en las redes de datos, como seguridad y básicamente eficiencia de la red. Se debe brindar confiabilidad y un acceso rápido a los servicios, y esto se lo puede hacer manipulando el ancho de banda otorgado a usuarios o aplicaciones dentro de cada red.

Siendo el ancho de banda un recurso limitado es necesario mantener control del mismo, controlar la calidad de servicio (QoS) y mantener la eficiencia de la red.

Con conocimiento en redes de datos, direccionamiento IP, subredes, puertos y un conocimiento de GNU/LINUX se puede diseñar e implementar una aplicación para controlar ancho de banda y que funcione como Firewall, para brindar un mejor servicio a los usuarios.

CAPÍTULO I

1 EL PROBLEMA

1.1 Planteamiento del Problema

En la actualidad el responsable de un servidor Linux debe poseer conocimientos técnicos y experiencia para administrar y controlar una red; sin embargo, no siempre se consiguen personas con estas características, por lo que se hace necesario buscar una forma sencilla para optimizar el tiempo de aprendizaje y conseguir mejoras en la calidad de servicio.

El presente trabajo analiza básicamente las siguientes variables:

Diseño e Implementación de Software Gráfico basado en Linux, como causa, y la Administración y Control de Ancho de Banda y Seguridades en Redes, como efecto.

La relación entre las variables anteriormente citadas es importante, en tanto sistema ofrezca una interfaz gráfica amigable, esto incide de forma positiva en la administración y control del ancho de banda en una red.

Un administrador de red no puede asignarle el mismo ancho de banda en áreas como desarrollo o gerencia y en atención al cliente.

Por ejemplo, es una subutilización de recursos el que un empleado, descargue archivos tipo mp3 y sature la red para que luego estos archivos sean enviados por correo electrónico corporativo a otros miembros de la empresa, mientras el gerente participa de una videoconferencia entre cortada y de mala calidad.

LINUX, permite elegir entre algunas versiones, se debe seleccionar la más estable y la que mejor se adapte a las necesidades de la red. Existen distribuciones como Red Hat, Fedora, Debian, etc. Actualmente muchos administradores de red utilizan los sistemas operativos basados en Linux por ser estables, confiables y además por la seguridad de archivos que es menos vulnerable que el sistema operativo Windows.

Con un sistema controlador de ancho de banda, con Firewall, que corra bajo una plataforma Linux, se puede mejorar la operación de la red y efectuar tareas como asignación diferenciada del ancho de banda, bloqueo total de clientes no autorizados, identificación automática de clientes por su IP y MAC, etc. sin que se torne lenta por aplicaciones indebidas. La interfaz de este controlador, debe presentar un entorno lógico y sobretodo fácil de manipular por cualquier persona, ya sea un administrador de red o personas con conocimientos básicos en redes.

1.2 Formulación del Problema

¿Cómo incide el diseño e Implementación de Software Gráfico basado en Linux para la Administración y Control de Ancho de Banda y Seguridades en Redes?

1.3 Objetivos

1.3.1 General

Desarrollar el software con una interfaz gráfica basada en Web, que facilite el control de ancho de banda y seguridades en una red.

1.3.2 Específicos

Analizar todas las herramientas posibles que GNU/LINUX ofrece para limitar ancho de banda.

Diseñar una Interfaz web de administración, fácil de utilizar, diseñada de acuerdo a criterios profesionales de operación.

Implementar un Firewall para completar la labor del controlador de ancho de banda.

Monitorear la operación del sistema controlador de Ancho de Banda.

1.4 Justificación

Los Administradores de red en la actualidad, utilizan sistemas operativos basados en GNU/LINUX, esto es por la seguridad, fiabilidad y robustez, que brindan.

La persona que vaya a manejar el software que se va a diseñar puede ser informático pero no necesariamente ser un experto en Linux, el software va a ser tan fácil de manipular que cualquiera lo podría hacer.

Este elemento de red que actualmente es diseñado por algunos fabricantes como: Cisco, Dell, ZyXEL WatchGuard etc. es de costo muy elevado, algunos sobre los 3000 USD, puede ser implementado con un PC con características necesarias de memoria y procesador.

1.5 Alcance

El presente trabajo finaliza con la implementación de la aplicación que permite controlar el ancho de banda, facilitando al administrador el manejo de la red, además, este debe ir de la mano con un firewall.

Se pretende que el sistema sea de fácil manejo, lo cual se logrará utilizando un diseño amigable a través de tecnologías que permitan que el usuario interactúe con la máquina.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Redes de Datos

2.1.1. Descripción.

Las redes surgen como respuesta a la necesidad de compartir datos de forma rápida. Las PC's son herramientas potentes que pueden procesar y manipular rápidamente grandes cantidades de datos, pero no permiten que los usuarios compartan los datos de forma eficiente.

Antes de la aparición de las redes, los usuarios necesitaban imprimir sus documentos o copiar los archivos en un disco para que otras personas pudieran editarlos o utilizarlos. Si otras personas realizaban modificaciones en el documento, no existía un método fácil para combinar los cambios. A este sistema se lo conoce como "trabajo en un entorno independiente".

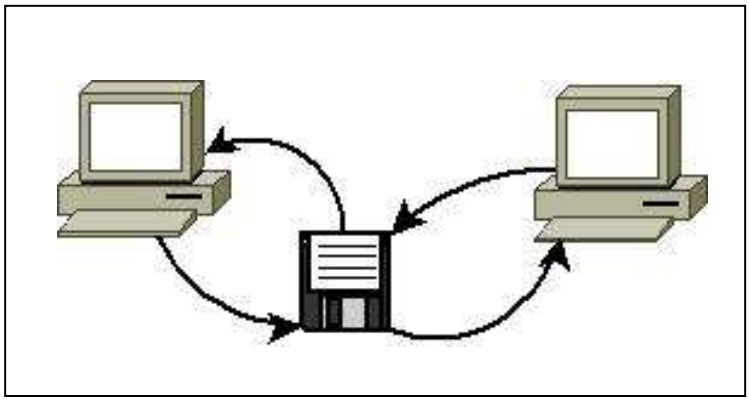


Figura 2-1 Información compartida por medios físicos [A]

Este sistema funciona bien en ciertas situaciones, pero resulta demasiado lento e ineficiente para cubrir las necesidades y expectativas de los usuarios informáticos de hoy en día. La cantidad de datos que se necesitan compartir y las distancias que deben cubrir los datos superan las posibilidades del intercambio de CD's, DVD's o memorias flash.

Si un equipo estuviera conectado a otros, entonces podría compartir datos con otros equipos, y enviar documentos a otras impresoras. Esta interconexión de equipos y otros dispositivos se llama **Red**.

Clasificación de las redes según su tamaño y extensión:

Redes LAN. Las redes de área local (Local Area Network) son redes de ordenadores cuya extensión es de entre 10 metros a 1 kilómetro. Son redes pequeñas, habituales en oficinas, colegios y empresas pequeñas. Las velocidades de transmisión típicas de LAN van de 10 a 1000 Mbps (Megabits por segundo).

Redes MAN. Las redes de área metropolitana (Metropolitan Area Network) son redes de computadoras de tamaño superior a una LAN, abarcando el tamaño de una ciudad. Se pueden encontrar en empresas y organizaciones que poseen distintas oficinas repartidas en una misma área metropolitana, por lo que, en su tamaño máximo, comprenden un área de unos 10 kilómetros.

Redes WAN. Las redes de área amplia (Wide Area Network) tienen un tamaño superior a una MAN. Consiste en una colección de redes LAN conectadas por una subred. Esta subred está formada por una serie de líneas de transmisión interconectadas por medio de routers. Su tamaño puede oscilar entre 100 y 1000 kilómetros.

Redes inalámbricas. Las redes inalámbricas son redes cuyos medios físicos no son cables, lo que las diferencia de las redes anteriores. Están basadas en la transmisión de datos mediante ondas de radio, microondas, satélites o infrarrojos.

RED	COBERTURA
LAN	10 metros a 1 kilómetro
MAN	10 kilómetros
WAN	Entre 100 y 1000 kilómetros
Inalámbricas	1 kilómetro o más

Cuadro 2-1 Clasificación de Redes [A]

2.1.2 Topologías

La topología es la ubicación física en la que se encuentran los nodos de una red, ya que a esta se la puede unir de varias formas, siendo la topología un factor fundamental que determina el rendimiento y la funcionalidad de la red.

Podemos distinguir dos aspectos diferentes a la hora de considerar una topología:

La topología física, que es la disposición real de las máquinas, dispositivos de red y cableado en la red.

La topología lógica, que es la forma en que las máquinas se comunican a través del medio físico. Los dos tipos más comunes de topologías lógicas son broadcast (Ethernet) y transmisión de tokens (Token Ring).

Modelos de topología física. Los principales modelos de topología son: **Topología en bus.-** La topología en bus tiene todos sus nodos conectados directamente a un enlace y no tiene ninguna otra conexión entre nodos. Físicamente cada máquina está conectado a un cable común, por lo que se pueden comunicar directamente, aunque la ruptura en un punto del cable hace que los hosts queden desconectados.

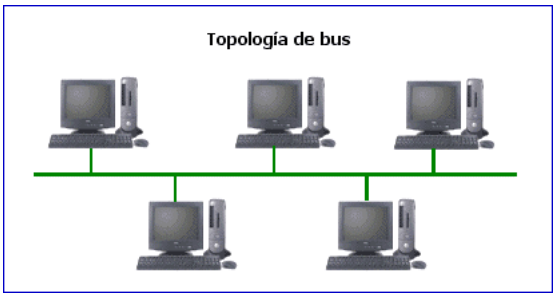


Figura 2-2 Topología de Bus [1]

La topología de bus permite que todos los dispositivos de la red puedan ver las señales de todos los dispositivos conectados a la misma, lo que puede ser ventajoso si se desea que todos los dispositivos obtengan esta información.

Sin embargo, puede representar una desventaja, ya que es común que se produzcan problemas de tráfico y colisiones, que se pueden eliminar segmentando la red en varias partes. Esta es la topología más común en pequeñas LAN.

Topología en anillo.- Una topología de anillo se compone de un solo anillo cerrado, formado por nodos y enlaces, en el que cada nodo está conectado solamente con los dos nodos adyacentes.

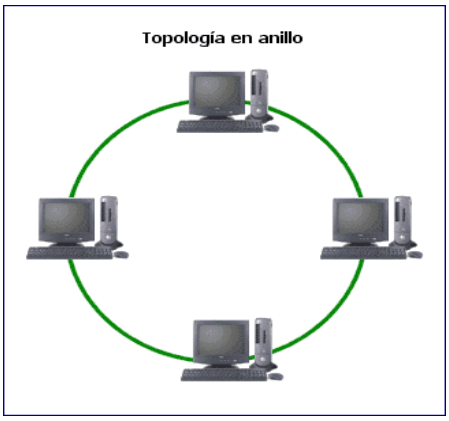


Figura 2-3 Título: Topología en Anillo [2]

Los dispositivos se conectan directamente entre sí por medio de cables. Para que la información pueda circular, cada estación debe transferir la información a la estación adyacente.

Topología en anillo doble.- Una topología en anillo doble consta de dos anillos concéntricos, donde cada host de la red está conectado a ambos anillos, aunque los dos anillos no están conectados directamente entre sí.

Es análoga a la topología de anillo, con la diferencia de que, para incrementar la confiabilidad y flexibilidad de la red, hay un segundo anillo redundante que conecta los mismos dispositivos. La topología de anillo doble actúa como si fueran dos anillos independientes, de los cuales se usa solamente uno por vez.

Topología en estrella.- La topología en estrella tiene un nodo central al que se conectan todos los equipos. Por este, que por lo general es un hub o un switch, pasa toda la información que circula por la red y la distribuye.

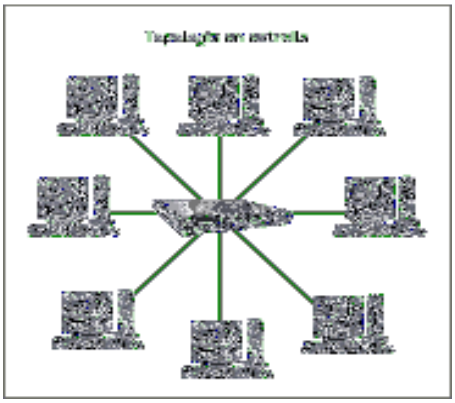


Figura 2-4 Título: Topología en Estrella [3]

La ventaja principal es que permite que todos los nodos se comuniquen entre sí de manera independiente. Su principal desventaja es que si el nodo central falla toda la red queda desconectada.

Topología en estrella extendida.- La topología en estrella extendida es igual a la topología en estrella, con la diferencia de que cada nodo que se conecta al nodo central también es el centro de otra estrella.

Generalmente el nodo central está ocupado por un hub o un switch, y los nodos secundarios por hubs. La topología en estrella extendida es sumamente jerárquica, y busca que la información se mantenga local.

Topología en árbol.- La topología en árbol es muy similar a la topología en estrella extendida, se diferencia de esta porque no tiene un nodo central. En su lugar posee un nodo de enlace troncal, generalmente ocupado por un hub o switch, desde el que se ramifican los demás nodos.

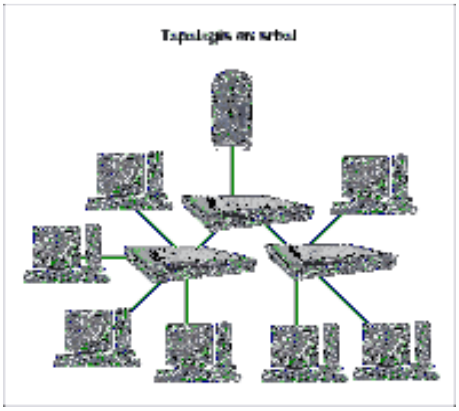


Figura 2-5 Título: Topología en Árbol [4]

El enlace troncal es un cable con varias capas de ramificaciones, y el flujo de información es jerárquico. Conectado en el otro extremo al enlace troncal generalmente se encuentra un host servidor.

Topología en malla completa.- En una topología de malla completa, cada nodo se enlaza directamente con los demás nodos.

La ventaja de esta topología es que como todos los nodos se conectan físicamente a los demás crean una conexión redundante, si algún enlace deja de funcionar la información puede circular a través de cualquier cantidad de enlaces hasta llegar a destino. Además, esta topología permite que la información circule por varias rutas a través de la red.

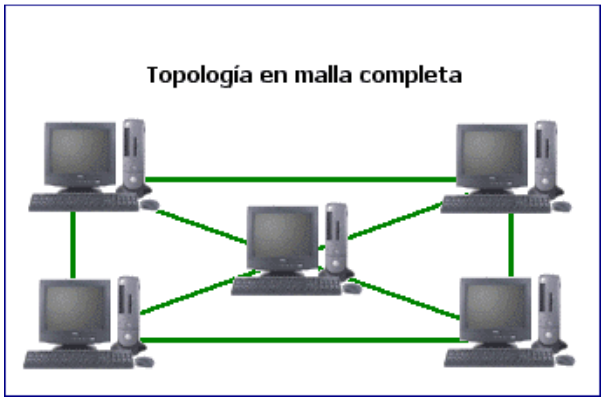


Figura 2-6 Título: Topología en Malla Completa [5]

La principal desventaja física de esta topología es que solo funciona con pocos nodos, ya que de lo contrario la cantidad de medios (cables) y la cantidad de conexiones se torna abrumadora.

Topología de red celular.- La topología celular está compuesta por áreas circulares o hexagonales, cada una de las cuales tiene un nodo individual en el centro.

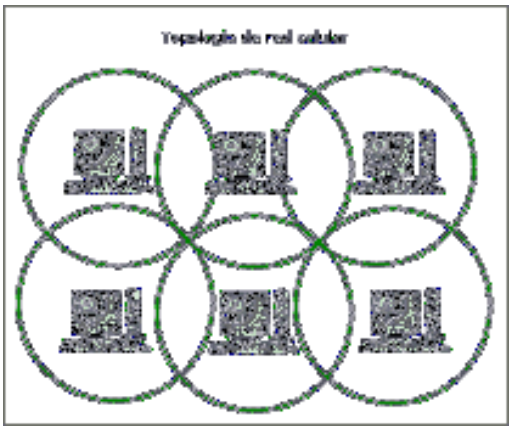
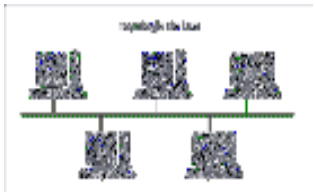
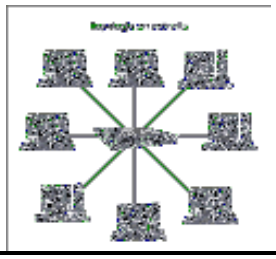
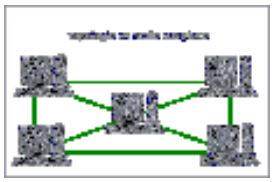
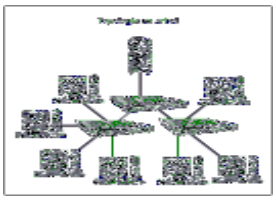
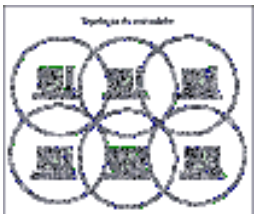


Figura 2-7 Título: Topología de Red Celular [6]

La topología celular es un área geográfica dividida en regiones (celdas) para los fines de la tecnología inalámbrica. En esta tecnología no existen enlaces físicos; sólo hay ondas electromagnéticas. La ventaja obvia de una topología celular (inalámbrica) es que no existe ningún medio tangible aparte de la atmósfera terrestre o el del vacío del espacio exterior (y los satélites). Las desventajas son que las señales se encuentran presentes en cualquier lugar de la celda y, de ese modo, pueden sufrir disturbios y violaciones de seguridad. Como norma, las topologías basadas en celdas se integran con otras topologías, ya sea que usen la atmósfera o los satélites.

Topología irregular En este tipo de topología no existe un patrón obvio de enlaces y nodos. El cableado no sigue un modelo determinado. Las redes que se encuentran en las primeras etapas de construcción, o se encuentran mal planificadas, a menudo se conectan de esta manera.

Topologías
 
 
 

Cuadro 2-2 Título: Topologías de Red [A]

Métodos de control de acceso al medio:

CSMA/CD: estas siglas corresponden a **Carrier Sense Multiple Access with Collision Detection** ("Acceso Múltiple con Escucha de Portadora y Detección de Colisiones"), es una técnica utilizada en redes Ethernet para mejorar sus prestaciones. En este método los dispositivos de red que tienen datos para transmitir, funcionan en el modo "escuchar antes de transmitir".

Esto significa que cuando un nodo desea enviar datos, primero debe determinar si los medios de red están ocupados, si alguien esta transmitiendo espera a que termine, caso contrario transmite y se queda escuchando en caso de colisiones.

Las colisiones existen cuando el canal esta libre y dos o mas maquinas comienzan a transmitir. Cuando existen colisiones las maquinas vuelven a transmitir.

Token Ring: el acceso al medio es determinista, a diferencia del CSMA/CD, la estación se conecta al anillo por una unidad de interfaz (RIU), cada RIU es responsable de controlar el paso de los datos por ella, así como de regenerar la transmisión y pasarla a la estación siguiente. Se usa en redes de área local con o sin prioridad, el token pasa de estación en estación en forma cíclica, inicialmente en estado desocupado. Cada estación cuando tiene el token (en este momento la estación controla el anillo), si quiere transmitir cambia su estado a ocupado, agregando los datos atrás y lo pone en la red, caso contrario pasa el token a la estación siguiente. Cuando el token pasa de nuevo por la estación que transmitió, saca los datos, lo pone en desocupado y lo regresa a la red.

Token Bus: este método es análogo a Token Ring, pero en lugar de ser utilizado por topologías en anillo, esta diseñado para topologías en bus. Se usa un token (una trama de datos) que pasa de estación en estación en

forma cíclica, es decir forma un anillo lógico. Cuando una estación tiene el token, tiene el derecho exclusivo del bus para transmitir o recibir datos por un tiempo determinado y luego pasa el token a otra estación, previamente designada. Las otras estaciones no pueden transmitir sin el token, sólo pueden escuchar y esperar su turno. Esto soluciona el problema de colisiones que tiene el CSMA/CD.

2.1.3 *Tecnologías de Red*

Modelo OSI

El modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open System Interconection) proporciona a los fabricantes un conjunto de estándares que aseguran una mayor compatibilidad e interoperabilidad entre los distintos tipos de tecnología producidos por las diferentes empresas a nivel mundial.

Las ventajas que este modelo proporciona son las siguientes:

- Divide la comunicación de red en partes más pequeñas y sencillas.

- Normaliza los componentes de red para permitir el desarrollo y el soporte de los productos de diferentes fabricantes.

- Permite a los distintos tipos de hardware y software de red comunicarse entre sí.

- Impide que los cambios en una capa puedan afectar las demás capas, para que se puedan desarrollar con más rapidez.
- Divide la comunicación de red en partes más pequeñas para simplificar el aprendizaje.

Este modelo está dividido en siete capas:

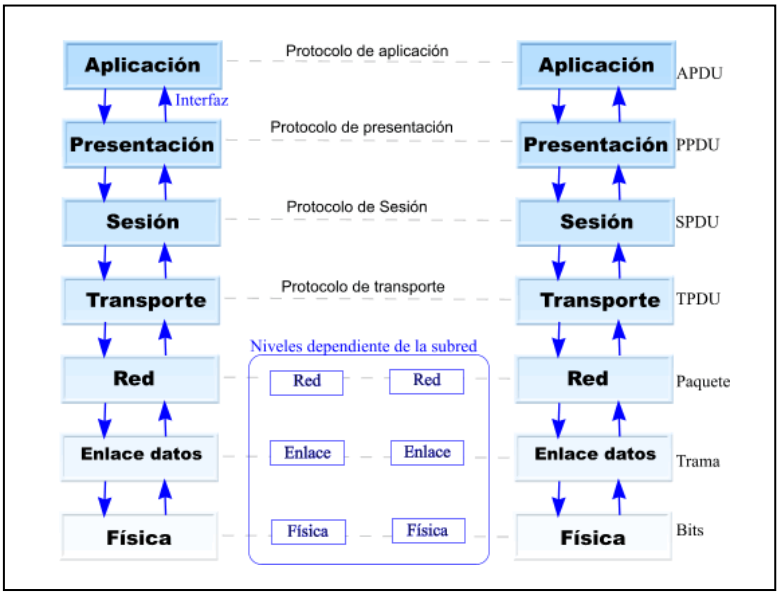


Figura 2-8 Título: Modelo OSI [7]

Capa Física (Physical Layer)

Esta capa se encarga de transmitir los bits de información por el medio utilizado para la transmisión. Se encarga de las propiedades físicas y características eléctricas de los diversos componentes.

Esta capa especifica el medio, los niveles de voltaje, características eléctricas, método de modulación, multiplexación, aspectos mecánicos y eléctricos de la interfaz de red y topología física.

Capa de Enlace (Data Link Layer)

Esta capa traslada los mensajes hacia y desde la capa física a la capa de red. Especifica como se encuentran organizados los datos cuando se transmiten en un medio particular.

Esta capa se encarga de definir una estructura de transferencia, ensamblar y reensamblar mensajes provenientes del nivel de red y enviarlos en tramas a través del medio físico. Además detecta y corrige errores provenientes del medio físico. También implementa el mecanismo de acceso al medio en medios compartidos.

Ejemplo: ATM, Ethernet, Frame Relay, HDLC, PPP, Token Ring, Wi-Fi.

Capa de Red (Network Layer)

Esta capa se ocupa de la transmisión de los datagramas (paquetes) y de encaminar cada uno en la dirección adecuada ("Routing").

Ejemplo: AppleTalk**,** IP, IPX, NetBEUI, X.25

Capa de Transporte (Transport layer).

Esta capa actúa como un puente entre las tres capas inferiores totalmente orientados a las comunicaciones y las tres capas superiores totalmente orientados a el procesamiento. Además, garantiza una entrega confiable de la información.

Asegura que la llegada de datos del nivel de red encuentra las características de transmisión y calidad de servicio requerido por la capa de Sesión.

Ejemplo: SCTP, SPX, TCP, UDP

Capa de Sesión (Sesion layer).

La capa de sesión establece, administra y termina las sesiones entre aplicaciones. Incluyendo la resincronización de dos computadoras que están manteniendo una sesión.

Ejemplo: NetBIOS

Capa de Presentación (Presentation Layer)

Define la manera en que se representan los datos (sintaxis de la información).

Formatos como: MPEG, JPEG, XDR, ASCII, EBCDIC.

Mecanismos para manejar compresión.

Encriptación de datos.

Es el nivel clave para el sistema de seguridad del modelo OSI.

Ejemplos: ASN.1, MIME, SSL/TLS, XML.

Capa de Aplicación (Aplication Layer)

Define estándares a nivel aplicativo de acuerdo a los diferentes tipos de servicio: emulación de terminal, transferencia de archivos, navegación en Internet, correo electrónico, unidad de transferencia (stream).

Ejemplos: DNS, FTP, HTTP, IMAP, IRC, NFS, NNTP, NTP, POP3, SMB/CIFS, SMTP, SNMP, SSH, Telnet, SIP.

2.1.4 Elementos de una Red

En una red de computadoras podemos hablar tanto de hardware como de software. En el hardware se encontramos: estaciones de trabajo, servidores, tarjeta de red, cableado y equipos de conectividad. En el software se encuentra el sistema operativo de red (Network Operating System, NOS).

Estaciones de trabajo

Cada host conectado a la red puede seguir trabajando independientemente con sus propios procesos, pero al mismo tiempo se convierte en una estación de red con acceso al servidor.

Servidores

Los servidores son aquellas computadoras que son capaces de compartir sus recursos con otras. Los recursos compartidos pueden ser impresoras, unidades de disco, CD-ROM, directorios en disco duro e incluso archivos individuales. A los servidores se les da su nombre dependiendo del recurso que comparten. Por ejemplo son: servidor de discos, servidor de archivos, servidor de archivos distribuido, servidor de terminales, servidor de impresoras, servidor de discos compactos, servidor web y servidor de correo.

Tarjeta de Interfaz de Red

Cada máquina o host debe tener instalada una tarjeta de interfaz de red o NIC (Network Interface Card). En la mayoría de los casos, la tarjeta se adapta en la ranura de expansión de la computadora, aunque algunas son unidades externas que se conectan a ésta a través de un puerto serial o paralelo. Las tarjetas de interfaz también pueden utilizarse en minicomputadoras y mainframes. La tarjeta de interfaz se encarga de

obtener la información de la PC, la convierte al formato adecuado y la envía a través del cable a otra tarjeta de interfaz de la red local. Esta tarjeta recibe la información y la traduce para que la PC pueda entender.

Las funciones de la tarjeta son: comunicaciones de host a tarjeta, buffering, formación de paquetes, conversión serial a paralelo, codificación y decodificación, acceso al cable, saludo, transmisión y recepción.

Cableado

Una red debe tener un sistema de cableado que conecte las estaciones de trabajo individuales con los servidores de archivos y otros periféricos. Existen muchos tipos de cableado, con una gran variedad en cuanto al costo y a la capacidad.

Cable de par trenzado: este cable es económico y es el más utilizado como medio de red.

Cable coaxial: Es tan fácil de instalar y mantener como el cable de par trenzado, y es el medio que se prefiere para redes de más de 100 metros.

Cable de fibra óptica: Tiene mayor velocidad de transmisión que los anteriores, es inmune a la interferencia de frecuencias de radio y capaz de enviar señales a distancias considerables sin perder su fuerza. Tiene un costo mayor.

Equipo de conectividad

Por lo general, para redes pequeñas, la longitud del cable no es limitante para su desempeño; pero si la red crece, tal vez llegue a necesitarse una mayor extensión de la longitud de cable o exceder la cantidad de nodos especificada. Existen varios dispositivos que extienden la longitud de la red, donde cada uno tiene un propósito específico. Sin embargo, muchos dispositivos incorporan las características de otro tipo de dispositivo para aumentar la flexibilidad y disminuir el valor.

Hubs o concentradores es un equipo de redes que permite conectar entre sí otros equipos y retransmite los paquetes que recibe desde cualquiera de ellos a todos los demás.

Repetidores: Un repetidor es un dispositivo que permite extender la longitud de la red; amplifica y retransmite la señal de red.

Puentes: Un puente es un dispositivo que conecta dos LAN separadas para crear lo que aparenta ser una sola LAN.

Ruteadores: Los ruteadores son similares a los puentes, sólo que operan a un nivel diferente. Requieren por lo general que cada red tenga el mismo sistema operativo de red, para poder conectar redes basadas en topologías lógicas completamente diferentes como Ethernet y Token Ring.

Compuertas: Una compuerta permite que los nodos de una red se comuniquen con tipos diferentes de red o con otros dispositivos. Podría tenerse, por ejemplo, una LAN que consista en computadoras compatibles con IBM y otra con Macintosh.

Sistema operativo de red

Después de cumplir todos los requerimientos de hardware, se necesita instalar un sistema operativo de red (Network Operating System, NOS), que administre y coordine todas las operaciones de dicha red. Los sistemas operativos de red tienen una gran variedad de formas y tamaños, debido a que cada organización que los emplea tiene diferentes necesidades. Algunos sistemas operativos se comportan excelentemente en redes pequeñas, así como otros se especializan en conectar muchas redes pequeñas en áreas bastante amplias.

Los servicios que el NOS realiza son:

Soporte para archivos: Esto es, crear, compartir, almacenar y recuperar archivos, actividades esenciales en que el NOS se especializa proporcionando un método rápido y seguro.

Comunicaciones: Se refiere a todo lo que se envía a través del cable. La comunicación se realiza cuando por ejemplo, alguien entra a la red, copia un archivo, envía correo electrónico, o imprime.

Servicios para el soporte de equipo: Aquí se incluyen todos los servicios especiales como impresiones, respaldos en cinta, detección de virus en la red, etc.

2.1.5 Administración de Redes

2.1.5.1 Introducción

El objetivo principal de la administración de red es en mantener operativa la red satisfaciendo las necesidades de los usuarios. La utilización de herramientas adecuadas permite realizar de forma centralizada la administración de múltiples redes de gran tamaño compuestos de cientos de servidores, puestos de trabajo y periféricos.

2.1.5.2 Importancia de la administración de redes

Las tareas de administración varían dependiendo, entre otras cosas, del número de usuarios a administrar, los tipos de periféricos conectados al conmutador, las conexiones de red y el nivel de seguridad necesaria.

Un administrador de sistema tiene que proporcionar a los usuarios del sistema un entorno eficiente, seguro y fiable.

La delegación de las responsabilidades de administración varía de un sistema a otro. En sistemas pequeños se asigna a un simple usuario la tarea de administrador. Si se trabaja en un entorno de red, la administración la realiza un administrador de red.

Todos los sistemas Linux tienen un sólo usuario que puede realizar cualquier operación en el computador denominado superusuario, con un nombre especial de entrada llamado root.

2.1.5.3 Herramientas

Normalmente las herramientas de administración de red forman un conjunto muy heterogéneo de aplicaciones proveniente de, por ejemplo, el sistema de gestión de red, el Help Desk, herramienta de los fabricantes de los dispositivos, herramientas autónomas e independientes. Además muchas de estas herramientas suelen tener APIs (Application Program Interface) que permiten el acceso por programación.

Hoy en día estas herramientas corren sobre diferentes S.O. y suelen tener la característica de disponer de un interface gráfico de usuario basado en ventanas.

2.2. GNU/Linux

2.2.1. *Breve Historia*

GNU/Linux (GNU con Linux) es la denominación defendida por Richard Stallman y otros para el sistema operativo que utiliza el kernel Linux en conjunto con las aplicaciones de sistema creadas por el proyecto GNU. Comúnmente este sistema operativo es denominado como Linux, aunque esta denominación no es correcta.

Desde 1984, Richard Stallman y voluntarios están intentando crear un sistema operativo libre con un funcionamiento similar al UNIX, recreando todos los componentes necesarios para tener un sistema operativo funcional que se convertiría en el sistema operativo GNU.

En el comienzo de los años 1990, después de seis años, GNU tenía muchas herramientas importantes listas, como compiladores, depuradores, intérpretes de órdenes etc, excepto por el componente central: el núcleo (GNU Hurd). El desarrollo de este núcleo fue mucho más duro de lo que se esperaba.

Afortunadamente, no hubo que esperar el desarrollo del núcleo, porque Linux ya estaba disponible. Cuando Linus Torvalds escribió Linux, rellenó la última laguna importante. La gente pudo entonces poner Linux junto al sistema GNU para obtener un sistema libre completo: una versión basada en Linux del sistema GNU; el sistema GNU/Linux, para abreviar.

GNU/Linux se distribuye bajo la ***GPL General Public License*** por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación ó trabajo derivado tiene que tener esta licencia.

El sistema ha sido diseñado y programado por muchos programadores alrededor del mundo. El núcleo del sistema sigue en continuo desarrollo bajo la coordinación de *Linus Torvalds*, la persona de la que partió la idea de este proyecto.

Hoy en día, grandes compañías, como IBM, SUN, HP, Novell y RedHat, entre otras muchas, aportan a Linux grandes ayudas tanto económicas como de código.

En la actualidad existen una diversidad de programas y aplicaciones que están disponibles para este sistema y la calidad de los mismos aumenta de una versión a otra. La gran mayoría de los mismos vienen acompañados del código.

Casas de software comercial distribuyen sus productos para GNU/Linux y la presencia del mismo en empresas aumenta constantemente por la excelente relación calidad-precio que se consigue con este.

2.2.2 Kernel

2.2.2.1 Definición

El kernel ó núcleo es el corazón de un sistema operativo. El término "núcleo" (en inglés *kernel*) propiamente dicho se refiere al software de sistema de bajo nivel que provee una capa de abstracción sobre el hardware, control de discos y sistema de archivos, multitarea, balance de carga, comunicación en red y medidas de seguridad.

Es el núcleo, el que asigna los recursos de la máquina a los otros programas para que se ejecuten. El núcleo es una parte esencial de todo sistema operativo, pero inútil por sí solo; sólo puede funcionar en el contexto de un sistema operativo completo.

2.2.2.2 Funciones

Las funciones más importantes del mismo, aunque no las únicas, son:

- Administración de la memoria para todos los programas y procesos en ejecución.
- Administración del tiempo de procesador que los programas y procesos en ejecución utilizan.
- Es el encargado de que se pueda acceder a los periféricos del ordenador de una manera cómoda.

2.2.2.3 Versiones

Hasta que empezó el desarrollo de la serie 2.6 del núcleo, existieron dos tipos de versiones del núcleo:

Versión de producción*:* La versión de producción, era la versión estable hasta el momento. Esta versión era el resultado final de las versiones de desarrollo o experimentales.

Cuando el equipo de desarrollo del núcleo experimental, decidía que tenía un núcleo estable y con la suficiente calidad, se lanzaba una nueva versión de producción ó estable. Esta versión era la que se debía utilizar para un uso normal del sistema, ya que eran las versiones consideradas más estables y libres de fallos en el momento de su lanzamiento.

Versión de desarrollo: Esta versión era experimental y era la que utilizaban los desarrolladores para programar, comprobar y verificar nuevas características, correcciones, etc. Estos núcleos solían ser inestables y no se debían usar sin saber lo que se hacía.

Como interpretar los números de las versiones de las ***series por debajo de la 2.6***:

Las versiones del núcleo se numeraban con 3 números, de la siguiente forma: AA.BB.CC

AA: Indicaba la serie/versión principal del núcleo. Solo han existido la 1 y 2. Este número cambiaba cuando la manera de funcionamiento del kernel había sufrido un cambio muy importante.

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

CC: Indicaba nuevas revisiones dentro de una versión, en las que lo único que se había modificado eran fallos de programación.

Por ejemplo:

Versión del núcleo 2.4.0: Núcleo de la serie 2 (AA=2), versión de producción 4 (BB=4 par), primera versión de la serie 2.4 (CC=0)

Con la ***serie 2.6*** del núcleo, el sistema de numeración así como el modelo de desarrollo han cambiado. Las versiones han pasado a numerarse con 4 dígitos y no existen versiones de producción y desarrollo.

Las versiones del núcleo se numeran hoy en día con 4 dígitos, de la siguiente forma: AA.BB.CC.DD.

AA: Indica la serie/versión principal del núcleo.

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

CC: Indica nuevas revisiones menores del núcleo. Cambia cuando nuevas características y drivers son soportados.

DD: Este digito cambia cuando se corrigen fallos de programación o fallos de seguridad dentro de una revisión.

Hoy en día se suele usar el núcleo distribuido con la distribución que el usuario utiliza. Son las distribuciones las encargadas de distribuir núcleos estables a sus usuarios y estos núcleos se basan en el núcleo ("vanilla") distribuido por Linus Torvalds y el equipo de programadores del núcleo.

2.2.3 Distribuciones

Una distribución GNU/Linux es un conjunto de aplicaciones que permiten brindar mejoras para instalar fácilmente este sistema. Son 'sabores' de Linux que, en general, se destacan por las herramientas para configuración y sistemas de paquetes de software a instalar.

Existen numerosas distribuciones Linux. Cada una de ellas puede incluir cualquier número de software adicional (libre o no), como algunos que facilitan la instalación del sistema y una enorme variedad de aplicaciones, entre ellos, entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etcétera.

La base de cada distribución incluye el núcleo Linux, con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos de software, como BSD.

Usualmente se utiliza la plataforma XFree86 o la Xorg para sostener interfaces gráficas.

Linux Standard Base (Fundación de estándares Linux) es una organización consagrada a desarrollar una cooperación estrecha entre diferentes distribuciones. El Filesystem Hierarchy Standard (Estándar jerárquico de sistema de ficheros) es una importante herramienta de la organización para lograr una cierta normalización oficial.

Existen varias distribuciones algunas de ellas son:

Distribuciones No Comerciales

- Aurox (basada en Red Hat Linux)
- BestLinux
- Debian (x86/PPC)
- CentOS (basada en Red Hat Enterprise Linux)
- Fedora Core (x86/PPC) (basada en Red Hat Linux)
- Gentoo Linux (x86/PPC)
- Gnoppix (basada en Ubuntu, antes en Debian, de tipo CD autónomo)
- Knoppix (basada en Debian, de tipo CD autónomo)
- Kubuntu (x86/PPC/x86-64) (Ubuntu con KDE)
- Mandriva Linux (x86/PPC/x86-64) (antes Mandrake Linux)
- Pardus (basada en Debian)
- ROCK Linux
- Slackware
- OpenSuSE
- Trinux (basada en Debian, de tipo CD autónomo)
- Trustix Secure Linux
- Ubuntu Linux (x86/PPC/x86-64) (basada en Debian)
- VectorLinux (basada en Slackware)
- White Box (basada en Red Hat Enterprise Linux)
- Jarro Negro (basada en Slackware), entre las mas conocidas.

Distribuciones No Comerciales Hispanoamericanas

- ASLinux Desktop (distribución para escritorios de descarga gratuita basada en Debian y KDE mantenida por la empresa andaluza Activa Sistemas)
- EduLinux (una distribución educativa chilena)
- Gobierno GDF/Linux (creada por la Delegación Tlalpan del Gobierno del Distrito Federal (México), basada en Fedora)
- Jarro Negro (creada por la Comunidad Linux UNAM Naucalpan CLUN, por estudiantes del Colegio de Ciencias y Humanidades plantel Naucalpan, basada en Slackware y Debian)
- GuadaLinex (x86/PPC) (impulsada por la Junta de Andalucía (España) basada en Ubuntu, antes en Debian)
- JuegaLinex (x86/PPC) (Hermana de Guadalinex, pero con muchos juegos)
- gnUAMix (patrocinada por la Universidad Autónoma de Madrid, basada en
- Debian y de tipo CD autónomo)
- Linedux (distribución educativa creada en Lima - Perú y basada en Debian)
- LinEspa (creada por el foro LinuxenEspañol, basada en Debian)
- LinEx (creada por la Junta de Extremadura (España),)
- Linuxin (basada en Debian GNU/Linux 3.0 (Woody) y realizada para novatos)

- LliureX (creada por la Generalitat Valenciana (España) y orientada al sistema educativo, basada en Knoppix. Soporta 2 idiomas: español y valenciano)
- LUC3M (distribución de la Universidad Carlos III de Madrid)
- Molinux (creada por la Comunidad Autónoma de Castilla-La Mancha (España), basada en Ubuntu)
- Musix GNU+Linux (100% Libre. Destinado a músicos, técnicos sonidistas y usuarios en general)
- Pequelin (distribución educativa para niños y jóvenes, basada en Knoppix)
- Ututo-e (distribución 100% libre creada en Argentina y basada en Gentoo)
- Tuquito (distribución creada en Tucuman - Argentina y basada en Debian)
- Kwort (distribución creada en Rosario - Argentina y basada en Slackware)

Distribuciones comerciales

- ASLinux Desktop (distribución para escritorios de descarga gratuita basada en Debian y KDE mantenida por la empresa andaluza Activa Sistemas)
- Caldera Linux

- Conectiva Linux (distribución hecha especialmente para América Latina. Soporta 3 idiomas: español, portugués e inglés)
- Corel Linux (basada en Debian)
- Linspire (basada en Debian) (antes Lindows)
- Lycoris Desktop/LX
- Mandriva
- Tumix GNu/linux distribución de linux basada en Debian y orientada al mercado latino comercializada en Perú, Uruguay, Chile y Argentina).
- Red Hat Enterprise Linux
- SUSE Linux (x86/PPC)
- Turbolinux
- Xandros (basada en Corel Linux e inspirada en Debian)
- Yellow Dog Linux (para PPC, basada en Fedora Core PPC)

En el cuadro 2-3 se puede apreciar una explicación más detallada de las distribuciones.

Distribución	Kernel	Ambiente
Alinex	2.6.22	GNOME
aLinux	2.6.12	KDE
ALT Linux	2.6.25	KDE, Xfce
Annvix	2.4.32	
Arch Linux	2.6.31.5[23]	Any
Archie	2.6.22[26]	Xfce
Ark Linux	2.6.22.3	KDE
Arudius	2.6.13	Fluxbox
Asianux	2.6.18	KDE
Aurox	2.6.9	KDE
BackTrack	2.6.21.5	KDE, Fluxbox
BLAG Linux and GNU	2.6.25.10[28]	GNOME
CentOS	2.6.18	GNOME
CrunchBang Linux	2.6.27	Openbox
CRUX	2.6.27.8	Openbox
Damn Small Linux	2.4.31	JWM
Debian	2.6.26	GNOME, KDE, LXDE or Xfce (depending on installation media)
DeLi Linux	2.4.32	IceWM
DeMuDi	2.6.12	GNOME
Dreamlinux	2.6.28.5[30]	Xfce
dyne:bolic	2.6.18	Xfce
Easy Peasy	2.6.30.5[31]	Gnome, Ubuntu Netbook Remix
Elive	2.6.15/2.6.18.2	Enlightenment
EnGarde Secure Linux	2.6.17	None
Fedora	2.6.31.5[32]	GNOME, KDE, Xfce
Finnix	2.6.22	None
Foresight Linux	2.6.27	GNOME
Frugalware	2.6.30	None
Gentoo	2.6.32	Any
gnuLinEx	2.6.16	GNOME

Cuadro 2-3 Distribuciones de Linux 1/3 [A]

Distribución	Kernel	Ambiente
GoboLinux	2.6.24.4	KDE
gNewSense	2.6.15.27	GNOME
Impi Linux	2.6.11.7	KDE
Kanotix	2.6.22	KDE
Knoppix	2.6.24.4	LXDE
Kurumin Linux	2.6.18	KDE
Linspire	2.6.14	KDE
Lunar Linux	2.6.26	None
Mandriva Linux	2.6.31.5[35]	GNOME,KDE
MEPIS	2.6.22.14	KDE
Musix GNU+Linux	2.6.16	IceWM
Mutagenix	2.6.18	KDE
NimbleX	2.6.11	KDE
Nitix	2.4.21	None
openSUSE	2.6.31.5	GNOME, KDE, Xfce
OpenWRT	2.6.27.10	None
Paipix	2.6.14	KDE
Pardus	2.6.25	KDE
Parsix	2.6.23.9	GNOME
PCLinuxOS	2.6.16	KDE
Pie Box Enterprise Linux	2.6.9	GNOME
Puppy Linux	2.6.25.16	JWM
QiLinux	2.6.17	KDE
Red Flag Linux	2.6.9	KDE
Red Hat Enterprise Linux	2.6.18[36]	GNOME
Rxart Desktop	2.6.11	KDE
Sabayon Linux	2.6.29.1	KDE
Satux	2.6.22	GNOME
Scientific Linux	2.6.18	GNOME
sidux	2.6.31	KDE 4.3.2

Cuadro 2-4 Distribuciones de Linux 2/3 [A]

Distribución	Kernel	Ambiente
Slackware	2.6.27.7	KDE
SLAX	2.6.24.4	KDE
SMS - Slack Mini Server	2.6.29.5	KDE
SliTaz GNU/Linux	2.6.25.5	Openbox
Source Mage GNU/Linux	2.6.27.10[38] (ISO) or any	None
SUSE Linux	2.6.27.19	GNOME, KDE, Xfce
Symphony OS	2.6.16	Mezzo
SYS	2.6.26.2 (iso) / 2.6.32.1 (repository [39], server [40])	KDE, GNOME 2.28, Xfce (selectable in kdm)
Ubuntu/Edubuntu	2.6.31[41] / 2.6.24 (LTS)	GNOME
Kubuntu	2.6.31[41] 2.6.24 (LTS)	KDE
XBMC Live	2.6.27	XBMC Media Center
Xubuntu	2.6.31[41] 2.6.24 (LTS)	Xfce
Ututo GNU/Linux	2.6.27	GNOME
VectorLinux	2.6.13	Fluxbox, IceWM, Xfce
Wolvix	2.6.27.7	XFCE,OpenBox
Distribución	Kernel	Ambiente
Xandros Desktop OS	2.6.15	KDE
Yoper	2.6	KDE
Zenwalk Linux	2.6.30.5	Xfce

Cuadro 2-5 Distribuciones de Linux 3/3 [A]

2.2.4 Aplicaciones

Son un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajo.

2.3 Firewall e IPtables

2.3.1 Firewall

2.3.1.1 Qué es?

Un firewall o cortafuegos es un sistema o grupo de sistemas que refuerza las políticas de control de acceso entre redes. El firewall puede estar implementado en software, como una aplicación especializada corriendo en un computador individual, o bien puede tratarse de un dispositivo especial dedicado a proteger uno o más computadores.

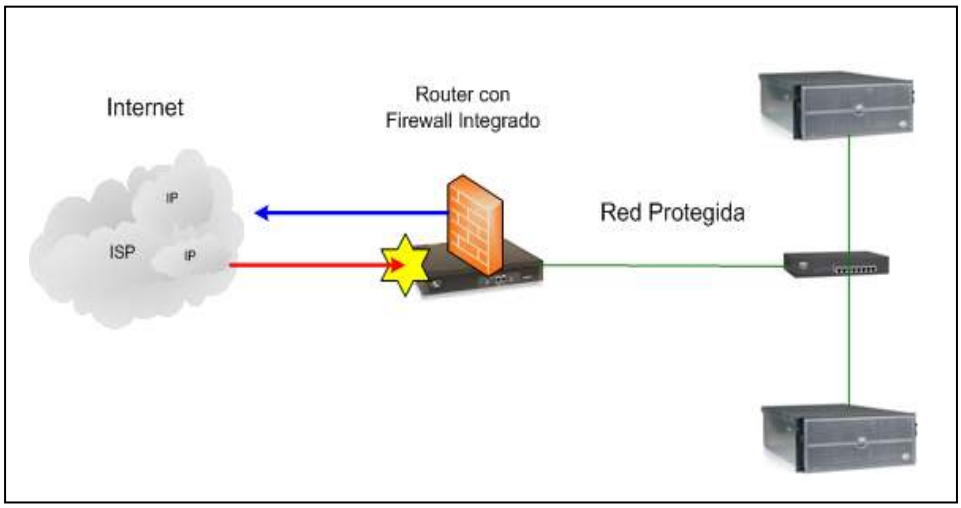


Figura 2-9 Firewall [8]

En general se utiliza, o bien la aplicación de firewall para proteger una máquina concreta conectada directamente a Internet (ya sea directa o por proveedor), o bien podemos colocar en la red una o varias máquinas dedicadas a esta función, de modo que protejan la red interna.

Técnicamente, la mejor solución es disponer de un computador con dos o más tarjetas de red que aíslen las diferentes redes conectadas. El software "firewall" se encarga de conectar los paquetes de las redes y determinar cuáles pueden pasar o no, y a qué red.

El firewall en general permite definir al administrador una serie de políticas de acceso (cuáles son las máquinas a las que se puede conectar o las que pueden recibir información y el tipo de información) por medio del control de los puertos TCP/UDP permitidos de entrada (*incomming*) o de salida (*outcomming*). Algunos firewalls vienen con políticas preconfiguradas; en algún caso sólo dicen si se quiere un nivel de seguridad alto, medio o bajo; otros permiten personalizar las opciones totalmente (máquinas, protocolos, puertos, etc.).

2.3.1.2 Tipos de Firewall

2.3.1.2.1 Firewall de capa de red

Funciona al nivel de red de la pila de protocolos (TCP/IP) como filtro de paquetes IP, no permitiendo que estos pasen al cortafuego a menos que se atengan a las reglas definidas por el administrador del cortafuego o aplicadas por defecto como en algunos sistemas inflexibles de cortafuego.

Una disposición más permisiva podría permitir que cualquier paquete pase el filtro mientras que no cumpla con ninguna regla negativa de rechazo

2.3.1.2.2 Firewall de capa de aplicación

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

2.3.1.3 Ventajas

Los firewalls en Internet administran los accesos posibles del Internet a la red privada. Sin un firewall, cada uno de los servidores propios del sistema se expone al ataque de otros servidores en el Internet. Esto significa que la seguridad en la red privada depende de la rigidez con que cada uno de los servidores cuenta.

Las principales ventajas son:

Protege de intrusiones, solamente entran a la red las personas autorizadas basadas en la política de la red en base a las configuraciones. El firewall permite al administrador de la red definir un “chek point” (embudo), manteniendo al margen los usuarios no-autorizados (tal como hackers, crakers y espías) prohibiendo potencialmente la entrada a vulnerar los servicios de la red, y proporcionar la protección para varios tipos de ataques posibles. Una de las ventajas clave de un firewall en Internet es que ayuda a simplificar los trabajos de administración, una vez que se consolida la seguridad en el firewall, es mejor distribuirla en cada uno de los servidores que integran la red privada.

Optimización de acceso, identifica los elementos de la red internos y optimiza que la comunicación entre ellos sea más directa si así se desea. Esto ayuda a reconfigurar rápida y fácilmente los parámetros de seguridad.

Protección de información privada. Permite el acceso solamente a quien tenga privilegios a la información de cierta área o sector de la red.

Protección contra virus. Evita que la red se vea infestada de nuevos virus que son liberados día a día.

El firewall ofrece un punto donde la seguridad puede ser monitoreada y si aparece alguna actividad sospechosa, este generara una alarma ante la posibilidad de que ocurra un ataque, o suceda algún problema en la transmisión de los datos. Esto se podrá notar al acceder la organización al Internet.

Finalmente, el firewall puede presentar los problemas que genera un único punto de falla simple. Enfatizando si este punto de falla se presenta en la conexión al Internet, aunque la red interna de la organización siga operando, únicamente el acceso al Internet está perdido.

2.3.2 IPtables

2.3.2.1 Descripción

IPtables es un sistema de firewall vinculado al kernel de Linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo. Al igual que el anterior sistema ipchains, un firewall de IPtables no es como un servidor que lo iniciamos o detenemos o que se pueda caer por un error de programación. IPtables está integrado con el kernel, es parte del sistema operativo. Para que IPtables funcione se aplican las reglas del script. Para ellos se ejecuta el comando IPtables, con el que añadimos, borramos, o creamos reglas.

Por ello un firewall de IPtables es un script de shell en el que se van ejecutando las reglas de firewall.

Las reglas de firewall están a nivel de kernel, y al kernel le llega un paquete y tiene que decidir qué hacer con él. El kernel lo que hace es, dependiendo si el paquete es para la propia maquina o para otra máquina, consultar las reglas de firewall y decide qué hacer con el paquete según indique el firewall. Este es el camino que seguiría un paquete en el kernel:

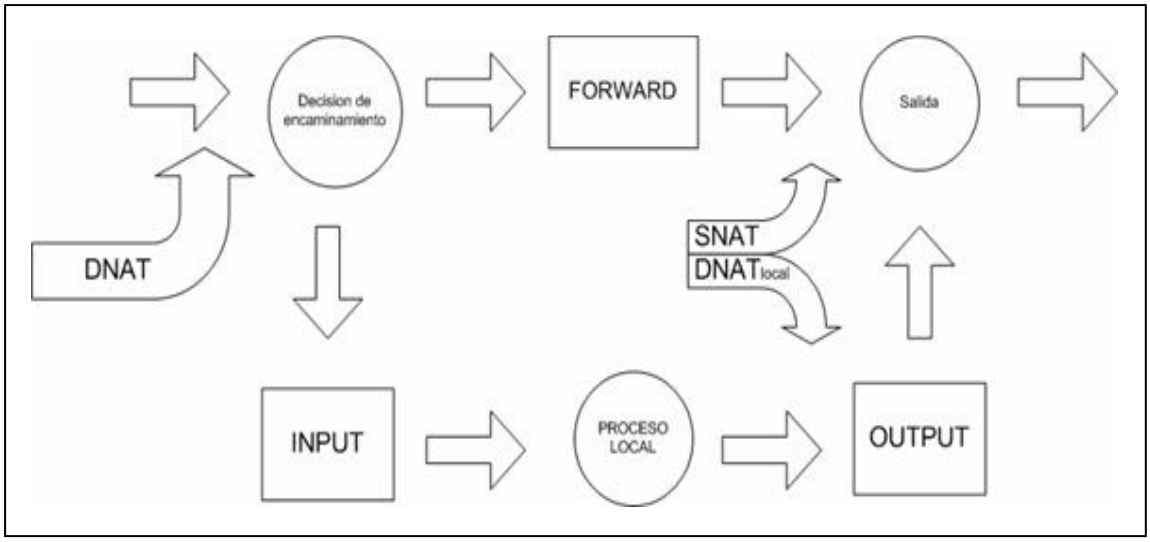


Figura 2-10 IPTables

Como se ve en el gráfico, básicamente se verifica si el paquete está destinado a la propia maquina o si va a otra. Para los paquetes (o datagramas, según el protocolo) que van a la propia maquina se aplican las reglas INPUT y OUTPUT, y para filtrar paquetes que van a otras redes o maquinas se aplica simplemente FORWARD.

INPUT, OUTPUT y FORWARD son los tres tipos de reglas de filtrado. Pero antes de aplicar esas reglas es posible aplicar reglas de NAT: estas se usan para hacer redirecciones de puertos o cambios en las IP's de origen y destino.

Incluso antes de las reglas de NAT se pueden crear reglas de tipo MANGLE, destinadas a modificar los paquetes; son reglas poco conocidas y es probable que no se usen.

Por tanto tenemos tres tipos de reglas en IPtables:

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

2.3.2.2 Especificación de reglas

Vamos a denominar cadena al conjunto de reglas asociados con un determinado tipo de filtro. IPtables también dispone de la posibilidad de que un usuario pueda definir sus propias cadenas y asignarle un nombre. Los nombres de cadenas predefinidos son INPUT, OUTPUT y FORWARD.

2.3.2.2.1 Acciones sobre un paquete

Las decisiones que puede tomar cada regla de un filtro de paquetes pueden ser, dejar pasar el paquete (ACCEPT), responderle al emisor que ese paquete no puede pasar (REJECT) o bien descartarlo como si no hubiera llegado (DROP o DENY).

La diferencia entre REJECT y DROP consiste en que REJECT indica que el servicio no está disponible (icmp destination port unrechable) evitando así demoras en la conexión; DROP no contesta nada por lo cual el sistema remoto no corta la conexión hasta que ha transcurrido el tiempo de espera de la contestación con la consiguiente relentización.

También tenemos la acción LOG que origina un registro de los paquetes que verifican la regla.

2.3.2.2.2 Características básicas de un paquete

Las características básicas de un paquete, que lo identificará son:

Dirección de origen

Indica quien es el emisor del paquete, de qué ordenador viene.

Lo podremos especificar con una dirección IP, un nombre de host o una dirección de red en formato CIDR (192.168.0.0/24) o en notación clásica (192.168.0.0/255.255.255.0). En IPtables podemos especificar la dirección origen con la opción "-s". Si en una regla omitimos la dirección origen equivale a poner 0/0 es decir cualquier dirección. Por ejemplo "-s 192.168.0.0/24" indicaría cualquier dirección con origen en la red de clase C (24 bits de red) 192.168.0.0. Si delante de la dirección añadimos "!" entonces hacemos referencia a cualquier dirección salvo la especificada, es decir, que no sea esa dirección.

Dirección de destino

Indica a quien va dirigido el paquete, a qué ordenador va. También lo podremos especificar con una dirección IP, un nombre de host o una dirección de red en formato CIDR (192.168.0.0/24) o en notación clásica (192.168.0.0/255.255.255.0). En IPtables podemos especificar la dirección destino con la opción "-d". Si en una regla omitimos la dirección origen equivale a poner 0/0 es decir cualquier dirección. Por ejemplo "-s 192.168.0.0/24" indicaría cualquier dirección con destino a la red de clase C, 192.168.0.0. Si delante de la dirección añadimos "!" entonces hacemos referencia a cualquier dirección salvo la especificada, es decir, que no sea esa dirección.

Protocolo

Podemos establecer filtros sobre protocolos concretos, será obligatorio si además especificamos algún puerto. La opción para especificar un protocolo es "-p" y los valores posibles son TDP, UDP e ICMP. El signo "!" antes del nombre del protocolo también se utiliza para negar.

Interfaz de entrada

Podemos especificar un dispositivo de entrada de red concreto con la opción "-i". Por ejemplo "-i eth0" indicaría un paquete que proviene de eth0. Se puede usar un "!". Evidentemente no podremos usar un interfaz de entrada con una regla de salida (OUTPUT).

Interfaz de salida

Podemos especificar un dispositivo de salida de red concreto con la opción "-o". Por ejemplo "-o eth0" indicaría un paquete que sale por eth0. Se puede usar un "!". Evidentemente no podremos usar un interfaz de salida con una regla de entrada (INTPUT).

Puerto origen

Mediante la opción "--sport" podemos especificar un puerto o un rango de puertos si los separamos por ":", por ejemplo [1024:65535] indicaría desde 1025 hasta 65535. Los puertos los podemos especificar por su número o también por el nombre asociado en el fichero /etc/services. Es necesario especificar -p TCP o -p UDP para poder especificar un puerto origen.

Puerto destino

Mediante la opción "--dport" podemos especificar un puerto o un rango de puertos. Las consideraciones son iguales que para el puerto origen.

Definición de reglas

Para definir una nueva regla es necesario detallar en el script de IPtables si tiene que insertar la nueva regla en alguna posición (-I numero) o bien simplemente añadirla al final (-A) y la descripción de la regla. Vemos algunos ejemplos:

Ejemplo 1:

iptables -A INPUT -s 192.168.0.0/24 -d 192.168.5.0 -p tcp --dport 80 -j ACCEPT

Se añade (-A) una regla de entrada (INPUT) que indica que todos los paquetes originados en la red 192.168.0.0 (-s 192.168.0.0/24) y dirigidos a la red 192.168.5.0 (-d 192.168.5.0) y dirigidos al puerto 80 (--dport 80) tienen que dejarse pasar (-j ACCEPT).

Ejemplo 2:

iptables -A FORWARD -s 192.168.0.7 -j REJECT

Se añade (-A) una regla de reenvío (FORWARD) que indica que todo el tráfico proveniente del ordenador 192.168.0.7 (-s 192.168.0.7) se rechaza (-j REJECT). Es decir estamos eliminando a este equipo la salida fuera de la red.

Ejemplo 3:

iptables -A OUTPUT -o eth0 -j ACCEPT

Permitimos cualquier salida por la interfaz de red eth0.

Ejemplo 4:

Vaciamos las reglas

iptables -F

iptables -X

```
iptables -t nat -F


## Establecemos predeterminada

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP


# permitimos el tráfico loopback

iptables -A INPUT -i lo -j ACCEPT

iptables -A OUTPUT -o lo -j ACCEPT


# Configuramos el acceso a nuestra IP

iptables -A INPUT -s 192.168.0.0/24 -j ACCEPT

iptables -A INPUT -s 0/0 -p tcp --sport 1:1024 -j ACCEPT
```

```
iptables -A INPUT -s 0/0 -p tcp --dport 1025:65535 ! --syn -j ACCEPT

iptables -A INPUT -s 0/0 -p udp --sport 1:1024 -j ACCEPT

iptables -A OUTPUT -d 192.168.0.0/24 -j ACCEPT

iptables -A OUTPUT -d 0/0  -p tcp --sport 1025:65535 -j ACCEPT



# El cortafuegos es también un servidor web

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT



# El cortafuegos es también un servidor smtp

iptables -A INPUT -p tcp --dport 25 -j ACCEPT

iptables -A OUTPUT -p tcp --sport 25 -j ACCEPT
```

Este es un ejemplo completo de IPtables en un firewall Linux.

2.4 Herramientas para el control de ancho de banda

2.4.1 Métodos de control de ancho de banda

2.4.1.1 Qué es QoS?

En general, en una red QoS se refiere a un conjunto de estándares y mecanismos que aseguran la calidad en la transmisión de los datos, siendo este conjunto una forma de medir los aspectos que afectan directamente la experiencia del usuario.

La QoS (*Quality of Service* o **C**alidad **d**e **S**ervicio) garantiza que se transmitirá cierta cantidad de datos en un tiempo dado (*throughput*). Una de las grandes ventajas de ATM (Modo de Transferencia Asíncrona) respecto de técnicas como el *Frame Relay* y *Fast Ethernet*, es que soporta niveles de QoS. Esto permite que los proveedores de servicios ATM garanticen a sus clientes que el retardo de extremo a extremo no excederá un nivel específico de tiempo. Además que en los servicios satelitales da una nueva perspectiva en la utilización del ancho de banda, dando prioridades a las aplicaciones de extremo a extremo con una serie de reglas.

2.4.1.2 Control de Tráfico en GNU/Linux

El sistema operativo GNU/Linux incorpora desde las versiones 2.2 y 2.4 de kernel, soporte para control de tráfico que aporta las funcionalidades básicas de disciplinas de colas, clases y filtros.

El sistema de control de tráfico en Linux se esquematiza en la siguiente figura:

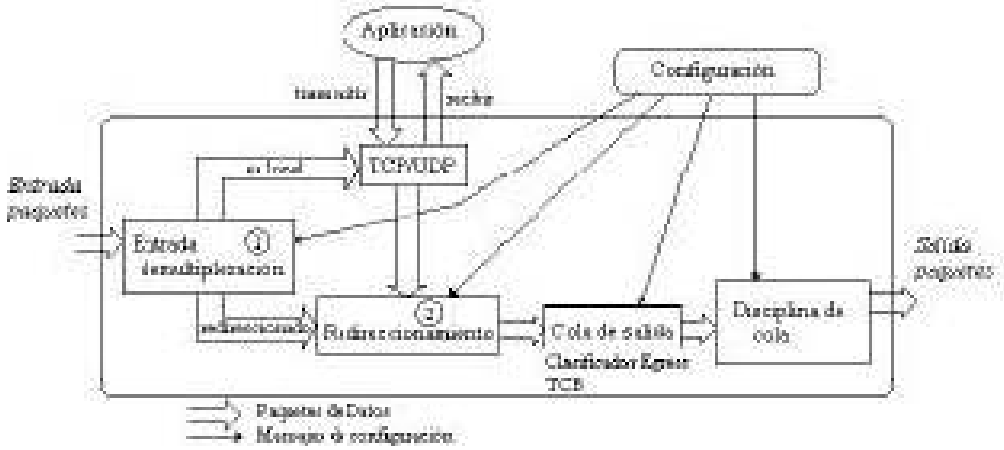


Figura 2-11 Procesos Iptables [9]

Los procesos del sistema operativo pueden transmitir y recibir paquetes haciendo uso de funcionalidades de red a nivel de kernel.

Una de estas funcionalidades se encargada del control de tráfico en Linux. La Figura 11 muestra su estructura de bloques.

La entrada demultiplexación (1) examina los paquetes de llegada y determina si el paquete es para la máquina local o hay que

encaminarlo a la siguiente máquina. Si el paquete es para la propia máquina, se envía a la capa superior para ser procesado. Si no, se pasa al bloque de redireccionamiento (2). Al bloque de redireccionamiento (2) también le llegan paquetes a transmitir por la red, generados por la propia máquina en capas superiores.

El redireccionamiento (2) observa las tablas de rutas, busca el siguiente salto, hace selección de la interfaz de salida adecuada para esa ruta y encapsula el paquete. Una vez hecho todo esto y tras determinar el nodo destino, almacena el paquete en la cola de salida, es decir, el buffer de salida adecuado a ese servicio, ruta o tipo de información, para ser transmitido en cuanto pueda, tras el acceso a la red (que depende de la topología de la red, por ejemplo, si es 802.2 Ethernet será por contienda con el protocolo CSMA/CD, y si es Token Ring esperará a coger el testigo).

El control de tráfico de Linux decide si el paquete se encola o se elimina (por ejemplo, porque las colas tengan longitud limitada y ya estén llenas, o porque el tráfico excede algún límite de velocidad de transmisión). El control de tráfico puede decidir en qué orden se envían los paquetes, dando por ejemplo prioridad a ciertos flujos frente a otros, y puede retardar la transmisión de los paquetes.

El control de tráfico que ofrece Linux se puede usar para construir combinaciones complejas de disciplinas de cola, clases y filtros útiles para aplicar disciplinas de QoS sobre los paquetes que se mandan a algún interfaz de salida.

En la base del funcionamiento del control de tráfico en Linux hay tres bloques principales:

***Qdisc*:** Disciplina de cola, lo que sería la cola que recoge los paquetes y los saca según la disciplina determinada. Existe por lo menos una en el interfaz de salida. Se puede especificar una disciplina de cola por cada clase final (en la que acaban los paquetes tras ser clasificados).

***Class*:** La clase determina de qué tipo de tráfico se trata. Más concretamente dice a qué tipo de servicio o a qué clasificación corresponde el paquete. Cada clase tiene sus propias características referentes al tipo de disciplina de cola que se le quiera asociar.

***Classifiers o filters*:** el filtro o el clasificador, dónde se determina el criterio para la discriminación de paquetes, bien sea por las direcciones origen y/o destino como por el tipo de información que lleva (según el puerto origen y/o destino), u otros campos de las cabeceras, con total flexibilidad.

Estos tres bloques son la base fundamental para entender en qué se basa un controlador de tráfico en Linux. La gestión del ancho de banda es jerárquica, es decir, que las clases siguen una jerarquía que guarda relación con el ancho de banda a determinar en cada clase.

Sólo puede existir una *qdisc* raíz por dispositivo. Esta *qdisc* se asocia al dispositivo de salida, la cual es dueña de todo el ancho de banda que éste ofrece.

2.4.1.3 Disciplinas de Colas

Existen dos tipos de disciplinas de colas: sin clase y con clase

2.4.1.3.1 Disciplinas de Colas Sin Clase

Estos son los procesos de encolar paquetes sencillos, pues sólo tienen la capacidad de reordenar, retrasar o descartar los paquetes que van llegando para ser enviados.

Existen tres procesos de encolar paquetes de este tipo y son:

- PFIFO_FAST
- TOKEN BUCKET FILTER (TBF)
- STOCHASTIC FAIRNESS QUEUEING (SFQ)

PFIFO_FAST.-

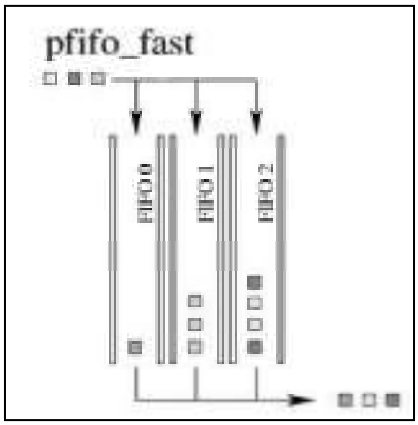


Figura 2-12 FIFO

Esta disciplina de colas está formada por tres bandas (bandas 0, 1 y 2). Dentro de cada banda los paquetes son enviados siguiendo una política FIFO (First In, First Out). Sin embargo, ningún paquete de la banda 1 es enviado mientras existan paquetes por enviar en la banda 0, y lo mismo ocurre para las bandas 2 y 1. Es decir, existe una prioridad definida entre dichas bandas, siendo la banda 0 la más prioritaria y la banda 2 la de menor prioridad. Para determinar los paquetes que van a cada banda se utiliza el campo TOS (Type Of Service) de la cabecera IP del mismo.

La disciplina de colas sin clases no tiene ninguna subdivisión interna en su estructura. Sin embargo, vemos como esta disciplina pfifo_fast sí la tiene, ya que esta posee tres bandas.

Lo importante es que una disciplina de colas sin clases no puede tener ninguna subdivisión interna de su estructura, susceptible de ser configurada por el usuario. Así pues, aunque la pfifo_fast tiene subdivisión interna, esta no puede ser modificada por el usuario.

TOKEN BUCKET FILTER (TBF).-

Este tipo de disciplina de colas es la que debemos escoger en el caso de que nuestra única necesidad sea limitar el ancho de banda de un determinado interfaz.

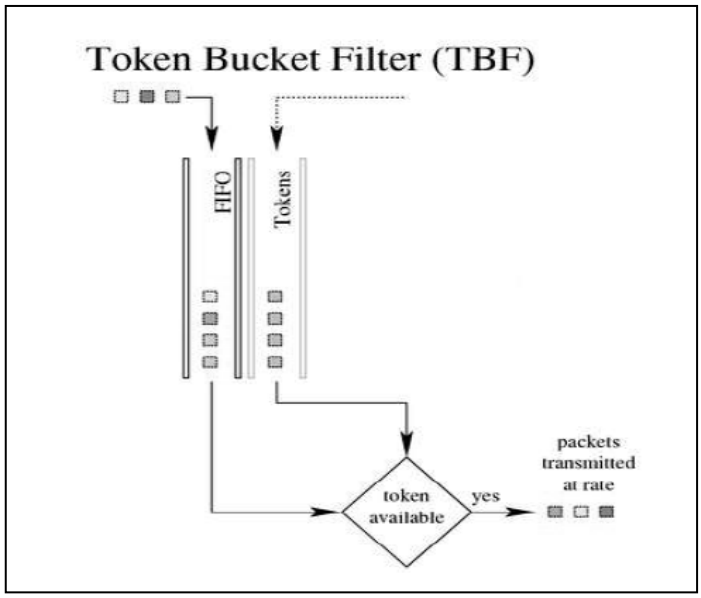


Figura 2-13 Cola TBF

El modelo de funcionamiento consiste en suponer que tenemos un buffer (bucket) al cual llegan los denominados 'tokens' a un ritmo constante. Estos tokens además serán los que utilizarán los paquetes IP para salir del interfaz de red.

Es como si cada paquete IP tuviese que esperar a una carretilla (token) que será la encargada de sacarlo del interfaz. Cada token que llega toma un paquete de datos entrante de la cola de datos y se elimina del bucket. Dependiendo de cuales sean los ritmos a los que entran los paquetes IP y los tokens podemos tener 3 posibles situaciones:

- Los paquetes IP llegan a mismo ritmo que los tokens. En este caso, cada paquete IP es asignado automáticamente a una de las carretillas que lo sacará del interfaz.

- Los paquetes IP llegan a un ritmo mayor que el de los tokens. En este caso, los paquetes IP tendrán que esperar durante un tiempo a que haya disponible una carretilla que les pueda sacar. Si esta situación se prolonga, parte de los paquetes IP que esperan, comenzarán a ser descartados, con lo cual limitamos el ancho de banda.

- Los paquetes IP llegan a un ritmo menor que el de los tokens. En este caso, cada paquete IP será asignado automáticamente a una carretilla que lo sacará del interfaz. Además los tokens, que no han sido utilizados para sacar ningún paquete IP, serán almacenados en el buffer (bucket) hasta alcanzar el límite del mismo.

De esta forma, si cambiara la tendencia y empezarán a llegar paquetes IP a un mayor ritmo, se podrían utilizar estos tokens almacenados para sacar, de manera instantánea, parte de esos paquetes IP que llegan.

STOCHASTIC FAIRNESS QUEUEING (SFQ).-

Este tipo de disciplina de colas intenta distribuir el ancho de banda de un determinado interfaz de red de la forma más justa posible.

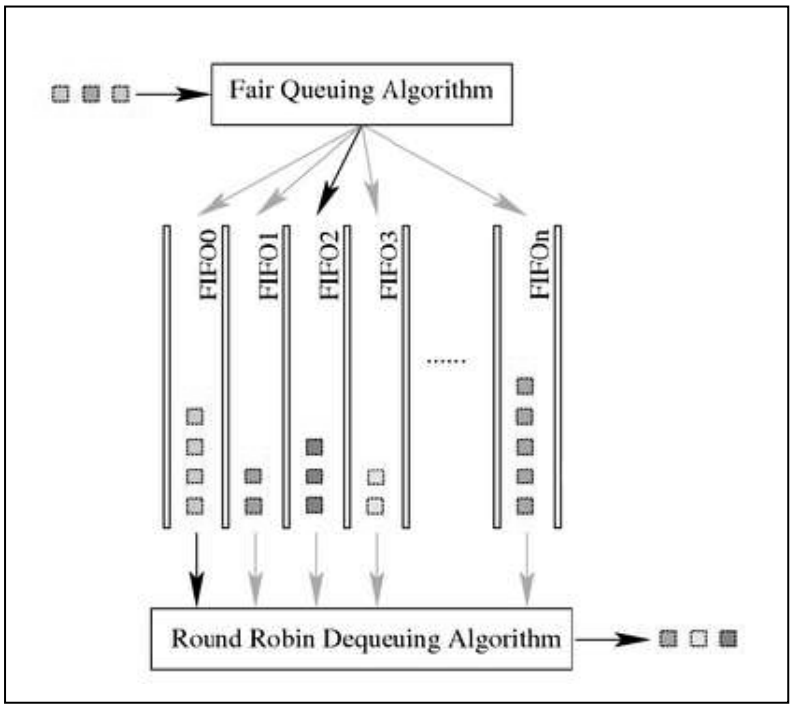


Figura 2-14 Cola SFQ

Para ello esta disciplina implementa una política de Round Robin[1] entre todos y cada uno de los flujos de comunicación establecidos en el interfaz, dando a cada uno la oportunidad de enviar sus paquetes por turnos. Un flujo de comunicación será cualquier sesión TCP o flujo UDP, y de esta forma lo que conseguimos es que ninguna comunicación impida al resto poder enviar parte de su información. Lógicamente esta disciplina de colas sólo tendrá sentido en aquellos interfaces que normalmente estén saturados y en los que no queramos que una determinada comunicación eclipse al resto.

2.4.1.3.2 Disciplinas de Colas Con Clase

Este tipo de disciplinas de colas se caracterizan por tener una subdivisión interna de su estructura, susceptible de ser configurada por el usuario, lo cual las hace muy útiles cuando tenemos diferentes tipos de tráfico que necesitan diferentes tratamientos.

[1] **Round robin** es un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional, normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento.

Cuando los paquetes IP llegan a una disciplina de este tipo, necesitan ser enviados a una de las clases que la componen, es decir, necesitan ser clasificados. Para realizar esta clasificación se consultan los filtros asociados a la disciplina de colas, los cuales devuelven un resultado que permite a la disciplina de colas determinar a qué clase debe ser enviado el paquete. Además, cada clase sabemos que tiene asociada una nueva disciplina de colas (con o sin clases), con lo que nuevas consultas a filtros pueden ser realizadas hasta conseguir clasificar el paquete completamente.

En Linux, cada interfaz de red tiene una disciplina de colas de salida (egress) llamada ‘root’, que es la primera de su estructura interna. Por defecto, si no se especifica otra cosa, esta disciplina es del tipo pfifo_fast. Además, a cada disciplina de colas le es asignado un ‘manejador’ que se utilizará en los comandos de configuración de dicha disciplina. Estos manejadores constan de dos partes, un ‘número mayor’ y un ‘número menor’ separados por ‘:’, así el manejador de la disciplina de colas ‘root’ es ‘1:0’. Normalmente el número menor del manejador de una disciplina de colas es siempre cero, y el número mayor de las clases adjuntas a una disciplina de colas debe coincidir con el número mayor de la misma.

Las principales disciplinas de cola con clases son las siguientes:

- PRIO
- Hierarquical Token Bucket (HTB)
- Class Based Queue (CBQ)

PRIO.-

Esta disciplina de colas es muy similar a la disciplina sin clases pfifo_fast, aunque es mucho más versátil y ofrece mayores posibilidades.

Por defecto esta disciplina de colas define tres clases, y cada una de ellas tiene asociada una nueva disciplina de colas con política FIFO. Entre las tres clases existe una prioridad de forma que mientras haya paquetes en la clase 1 no se envían paquetes de la clase 2, y lo mismo entre las clases 2 y 3. Vemos pues como hasta aquí trabaja de manera similar a la clase pfifo_fast. Sin embargo, la gran diferencia radica en dos factores:

1. En esta clase se puede definir los filtros que se crean necesarios, de forma que no estén limitados a hacer una clasificación de los paquetes en función del campo TOS, sino que se pueda hacer una clasificación compleja.

2. Aunque por defecto cada una de las tres clases asociadas a la disciplina PRIO tienen una disciplina con política FIFO, en realidad se puede definir la disciplina de colas que se necesiten. Por tanto, podría ser por ejemplo una nueva disciplina con clases, filtros asociados, etc.

Class Based Queueing (CBQ).-

Esta disciplina de colas fue la primera que se creó, y probablemente la más utilizada. De hecho, en muchos ámbitos aún se asocia el Control de Tráfico en Linux exclusivamente en referencia a este tipo de disciplina de colas. (En esta disciplina profundizaremos más, ya que en esta disciplina se basara el controlador de tráfico.)

Es un mecanismo de control de tráfico basado en compartir el ancho de banda de un enlace para así poder tener un mantenimiento de los recursos.

Tiene la característica de permitir compartir el enlace entre múltiples agentes, familias de protocolos o tipos de tráfico. Ofrece una estructura jerárquica por cada enlace, en la cual se encuentran las clases, que son las estructuras correspondientes a algún tipo de agregado de tráfico. La propia estructura jerárquica especifica la política deseada para ese enlace, expresada en porcentajes del ancho de banda, en periodos de congestión.

La reserva de este ancho de banda puede ser estática (mediante la asignación del administrador) o dinámica (por el uso de un protocolo de reserva de recursos).

Como objetivo principal de CBQ hay que destacar que una clase recibe por lo menos el ancho de banda que se le asoció, incluso en situación de congestión. Por lo tanto si es una clase con mucha demanda recibe como mínimo el ancho de banda asociado, pudiendo recibir más si las reglas así lo permitieran. Si una clase no tiene ancho de banda asociado, CBQ no garantiza ningún ancho de banda en caso de congestión. El ancho de banda asociado depende del mecanismo de planificación de paquetes del router. Una correcta selección de anchos de banda asociados hace más útil un controlador desde el punto de vista del usuario.

Es un mecanismo de control de tráfico basado en compartir el ancho de banda de un enlace para así poder tener un mantenimiento de los recursos.

Otra característica importante de CBQ es el préstamo de ancho de banda. La redistribución del ancho de banda que alguna clase no usa no es arbitraria. Tomando como ejemplo la configuración de la Figura 2-15, el nodo D debe tener por lo menos el 60% del ancho de banda asociado que tiene el nodo A, y el nodo E el 40%. La notación

de un mínimo ancho de banda se debe al hecho de poder permitir que una clase tome el ancho de banda sobrante de la clase superior de la que desciende. De modo que si el nodo D estuviera superando el ancho de banda asociado, es decir necesitara más del 60% del ancho de banda asociado de A, su padre, y éste se lo permitiera, tomaría de su ancho de banda asociado sobrante para cubrir la falta de ancho de banda de D. El nodo E tendría el ancho de banda sobrante del nodo A. Si la utilización de ancho de banda de los nodos E y D no superara el ancho de banda del nodo A, este ancho de banda sobrante se repartiría siguiendo los porcentajes con los nodos B y C si el nodo superior, enlace, así lo permitiera, como se observa en la figura 2-15.

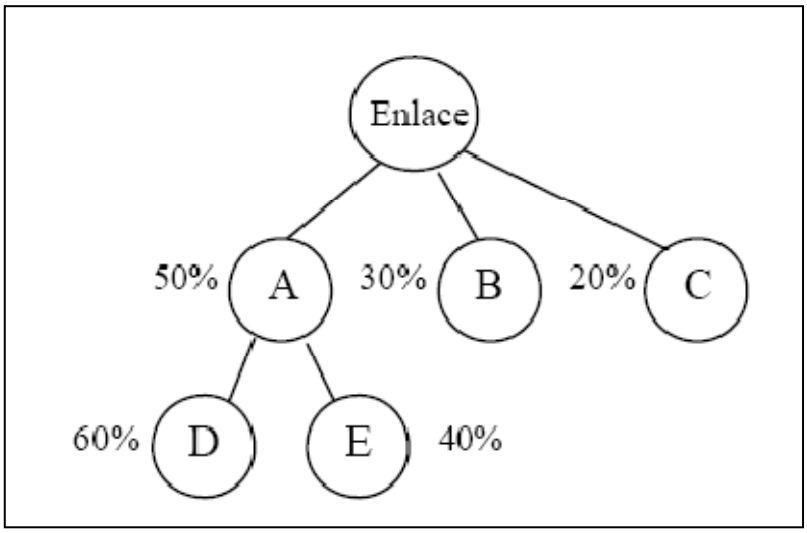


Figura 2-15 Ejemplo distribución CBQ

CBQ necesita también de los siguientes mecanismos:

- Clasificador de paquetes, para clasificar los paquetes que pasan por el router en la clase apropiada del enlace de salida. A este mecanismo CBQ no le especifica ningún requisito especial. El clasificador define la funcionalidad del planificador para el usuario, asociando los flujos con clases.

- Estimador del ancho de banda, es aquel que estima el ancho de banda que usa cada clase en un determinado intervalo de tiempo, para determinar si está recibiendo o no el ancho de banda asociado a esa clase. El intervalo de tiempo es un parámetro fundamental con el que se determina la precisión con la que el router cumple las relaciones jerárquicas del reparto del ancho de banda del enlace.

El número de parámetros que se utilizan para CBQ es elevado y a veces no se sabe cuál es exactamente su función, de ahí que en la mayoría de ocasiones debe ser la experiencia la que enseñe como utilizar estos parámetros.

Algunos de los más importantes son:

avpkt. Tamaño medio del paquete medido en bytes.

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

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

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

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

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

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

isolated/sharing. Una clase configurada con el parámetro ‘isolated’ no prestará nunca ancho de banda a sus hermanas. El

comportamiento contrario viene establecido por el parámetro 'sharing'. Por defecto, si no se indica lo contrario se supondrá que el 'sharing' está activo.

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

Hierarquical Token Bucket (HTB).-

La disciplina de colas CBQ es compleja. Básicamente CBQ es muy apropiada para casos en los que se dispone de un ancho de banda fijo que se quiera dividir en varios propósitos, dando a cada uno de ellos un ancho de banda garantizado, y con la posibilidad de prestar ancho de banda entre ellos.

HTB tiene una funcionalidad muy similar a la de CBQ, aunque su implementación es completamente distinta y su configuración menos compleja; Sin embargo, HTB aún no forma parte del kernel estándar de Linux y hay que parchearlo y recompilarlo para utilizarla.

2.4.2 Herramientas de monitoreo

2.4.2.1 IPTraf

IPTtraf es un programa basado en consola que proporciona estadísticas de red. El programa recolecta información de las conexiones TCP, las estadísticas y actividad de las interfaces, así como las caídas de tráfico TCP y UDP.

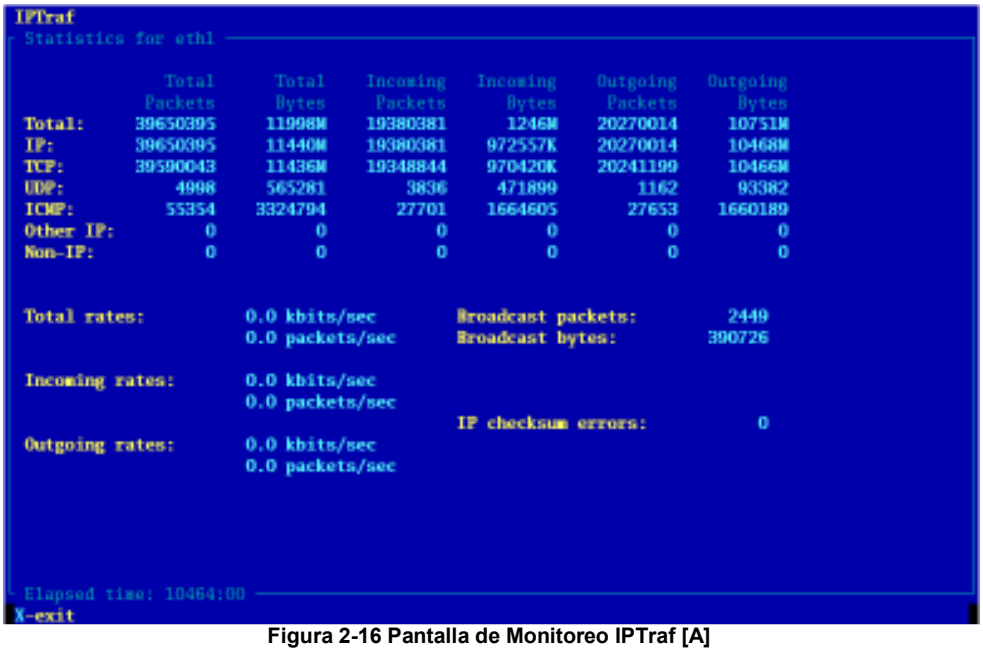


Figura 2-16 Pantalla de Monitoreo IPTraf [A]

Características

Además de un menú de opciones a pantalla completa, IPTraf posee las siguientes características:

- Monitor de tráfico IP que muestra información del tráfico de la red. Estadísticas generales de las Interfaces.

- Módulo de estadísticas de LAN que descubre hosts y muestra datos sobre su actividad.
- Monitor TCP, UDP que muestra la cuenta de los paquetes de red para las conexiones de los puertos de aplicaciones.
- Utiliza el "raw socket interface" que lleva el kernel permitiendo ser usado por un amplio rango de "tarjetas de red"

Protocolos reconocidos

IPTtraf admite la audición de múltiples protocolos: IP, TCP, UDP, ICMP, IGP, IGMP, IGRP, OSPF, ARP, RARP.

Interfaces admitidas

IPTraf admite una amplia gama de interfaces de red: Loopback local, interfaces Ethernet admitidas por GNU/Linux, interfaces FDDI admitidas por GNU/Linux, SLIP, asynchronous PPP, synchronous PPP over ISDN, ISDN con encapsulación Raw IP, ISDN con encapsulación Cisco HDLC, línea IP paralela.

2.4.2.2 MRTG

MRTG (Multi Router Traffic Grapher) es una herramienta, escrita por Tobias Oetiker y Dave Rand, para monitorizar la carga de tráfico sobre determinados nodos de una red. MRTG genera páginas

HTML que incluyen representaciones gráficas, en formato GIF, del tráfico registrado en un determinado nodo de la red.

MRTG consiste en un script en Perl que utiliza SNMP para obtener información de gestión sobre los nodos de la red y un programa en C para generar los registros de tráfico (logs) y crear representaciones gráficas de los datos recopilados. Estos gráficos se integran dentro de un documento en formato HTML.

Mediante MRTG es posible monitorear cualquier variable SNMP que se quiera, de marera que se puede configurar para monitorizar la carga de un sistema, las sesiones abiertas por los usuarios de un determinado equipo, disponibilidad de modems. MRTG permite generar gráficas con cuatro niveles de detalle por cada interfaz: tráfico registrado en las últimas 24 horas, la última semana, el último mes y gráfica anual. Además de generar una primera página con la representación del tráfico registrado diariamente a través de cada uno de los posibles interfaces de un router.

CAPÍTULO III

3 LA PROPUESTA

3.1 Desarrollo de la Aplicación

Hasta el capitulo anterior hemos verificado la teoría que será necesaria para desarrollar la aplicación, desde este capítulo iniciaremos con el desarrollo del mismo.

3.1.1 Instalación del Sistema Operativo

Hemos verificado que existen varias distribuciones de Linux y cualquiera de estas podría ser instalada para el desarrollo de la aplicación. La mayoría de administradores de red utilizan Red Hat Enterprise Linux debido a que este es el sistema operativo open source líder en el mundo, y aunque este se compone de software libre y código abierto, es una distribución comercial por lo que todos sus formatos binarios usables (CD-ROM o DVD-ROM) solamente son enviados a sus suscriptores pagados.

CentOS (Community ENTerprise Operating System), es un clon a nivel binario de la distribución Linux Red Hat Enterprise compilado por voluntarios a partir del código fuente liberado por Red Hat. Los desarrolladores de CentOS usan ese código fuente para crear un producto final que es muy similar al Red Hat Enterprise Linux y está libremente disponible para ser bajado y usado por el público, pero no es mantenido ni asistido por Red Hat. Es por esta razón que la distribución que utilizaré es CentOS.

Requisitos del Sistema

Hardware recomendado para operar:

- **Memoria RAM:** 64 MB (mínimo).
- **Espacio en Disco Duro:** 1024 MB (mínimo) - 2 GB (recomendado).
- **Procesador:** Intel Pentium I/II/III/IV/Celeron, AMD K6/II/III, AMD Turion, AMD Athlon/XP/MP.

Iniciamos la instalación, arrancamos desde el disco 1.

Figura 3-1 Pantalla Inicial de Instalación CentOS [A]

Proceso inicial y detección de Hardware: En la pantalla anterior, se presiona Enter, para iniciar el proceso para detección de Hardware y proceso de instalación a través de la consola gráfica, esta secuencia puede durar entre 10 o 15 segundos.

Verificación de medios (CD-ROM's): Posteriormente se presenta la opción para realizar una prueba de integridad sobre los CD-ROM's de instalación CentOS, esta prueba dura entre 10 y 15 minutos para los 4 CD's de instalación. Si no es necesario realizar esta prueba, seleccionar la opción Skip.

Figura 3-2 Pantalla de Verificación de Medios [A]

Lenguaje del Sistema: Es necesario escoger el idioma que requerimos para nuestro proceso de instalación.

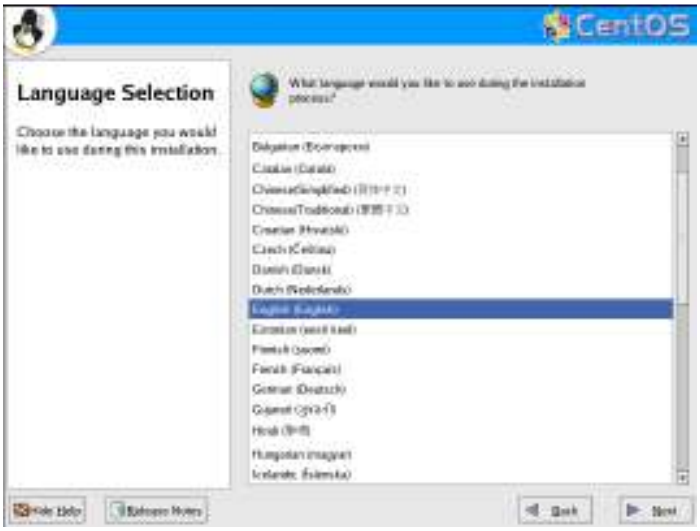


Figura 3-3 Pantalla de Selección del Idioma [A]

Tipo de Instalación: Se puede elegir entre 4 modalidades -- Escritorio Personal, Estación de Trabajo, Servidor o Personalizada -- cada una de éstas presenta una breve descripción de su funcionamiento. Se seleccionará la perzonalizada, para decidir que paquetes se debe instalar.

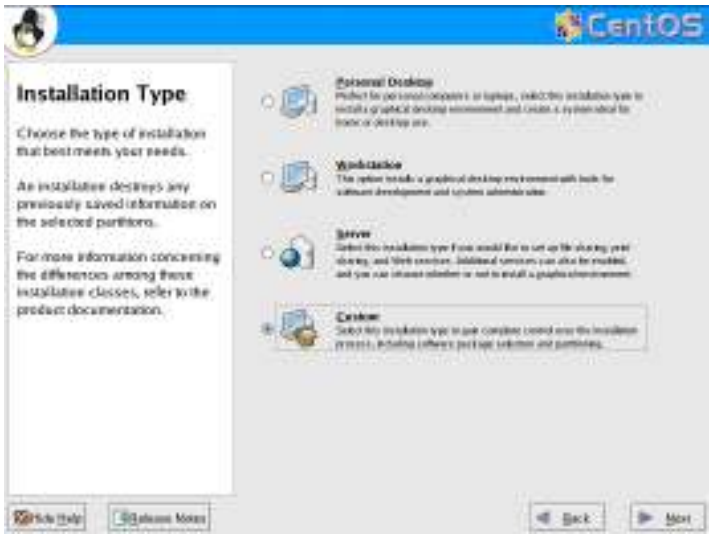


Figura 3-3 Pantalla de Tipo de Instalación

Partición de Disco Duro: Posteriormente, se debe realizar el particionamiento del disco duro, CentOS ofrece dos alternativas para llevar a cabo este proceso.

Particionamiento Automático: Como su nombre lo indica, CentOS realiza el particionamiento del disco duro a dimensiones pre-determinadas, sin embargo, esto implica generalmente que debe borrar toda la información existente en el disco duro.

Partición manual con Disk Druid: Para usuarios con amplio conocimiento del proceso de partición, pueden optar por hacer su propia distribución de espacio con esta opción.

Figura 3-4 Pantalla de Particionamiento de Disco

En este caso se selecciona el particiomaniento manual, dando los siguientes valores:

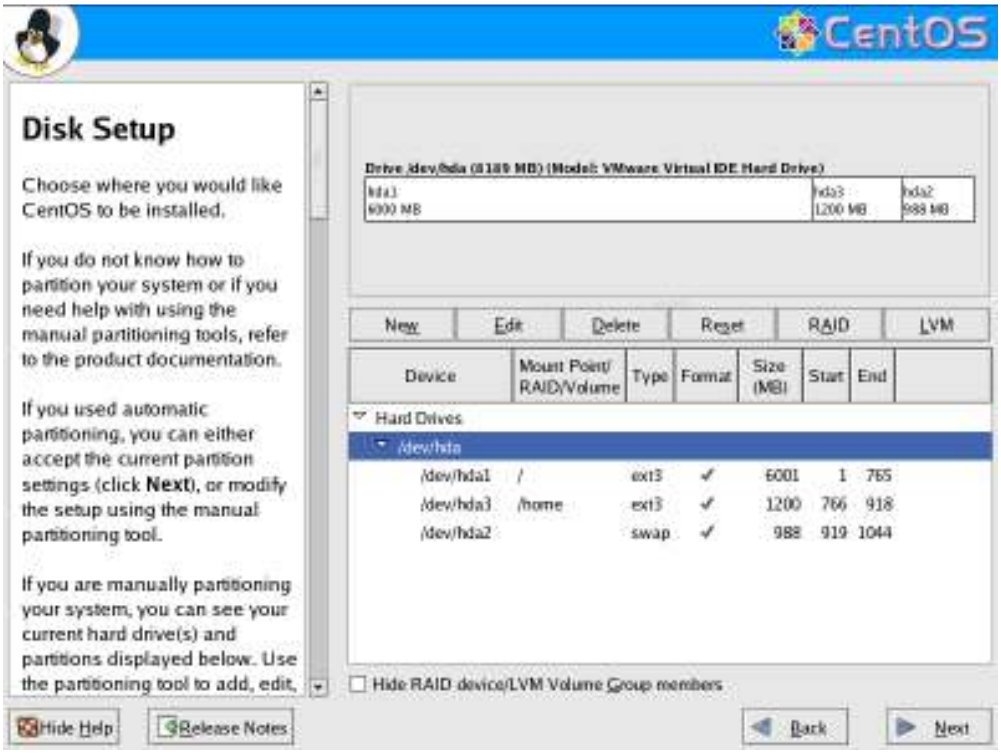


Figura 3-5 Pantalla de Particionamiento Manual [A]

Administrador de Arranque ("Boot Loader"): Posteriormente debe confirmar la instalación del administrador de arranque ("Boot Loader") GRUB; si CentOS es el único sistema operativo instalado en su equipo, este paso no debe ser de mayor importancia. Sin embargo, si posee más de un disco duro, o además de CentOS existirá otro sistema operativo, esta configuración tiene implicaciones en la manera que es inicializado el sistema.

Figura 3-6 Pantalla de Administrador de Arranque [A]

Configuración de Red: Seguido se debe indicar los parámetros para acceso a red, ya sea manualmente con información IP y DNS, o bien, indicando una configuración automática vía DHCP.

Para la aplicación se configuró el ETH0 como DHCP y el ETH1 con la IP 192.168.0.1 y mascara de red 255.255.255.0.

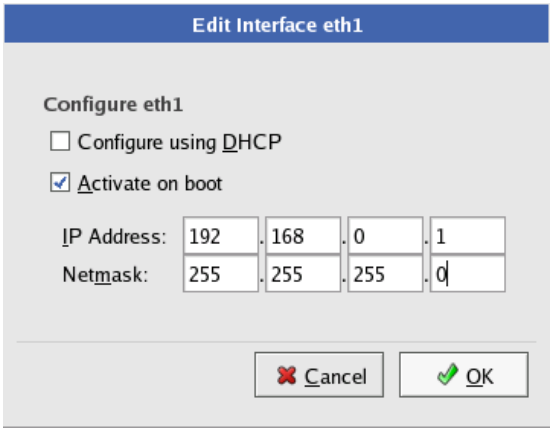


Figura 3-7 Pantalla de Configuración de Interfaz de Red [A]

Configuración Cortafuegos ("Firewall"): Aquí se debe especificar si instalará un mecanismo de "Firewall" para proteger su sistema. Si es así, también existe la opción de habilitar determinados servicios para que éstos no sean afectados por el "Firewall", tales como: SSH, Servidores Web, Servidores de Correo y FTP. Para que la aplicación funcione el SELinux debe estar deshabilitado.

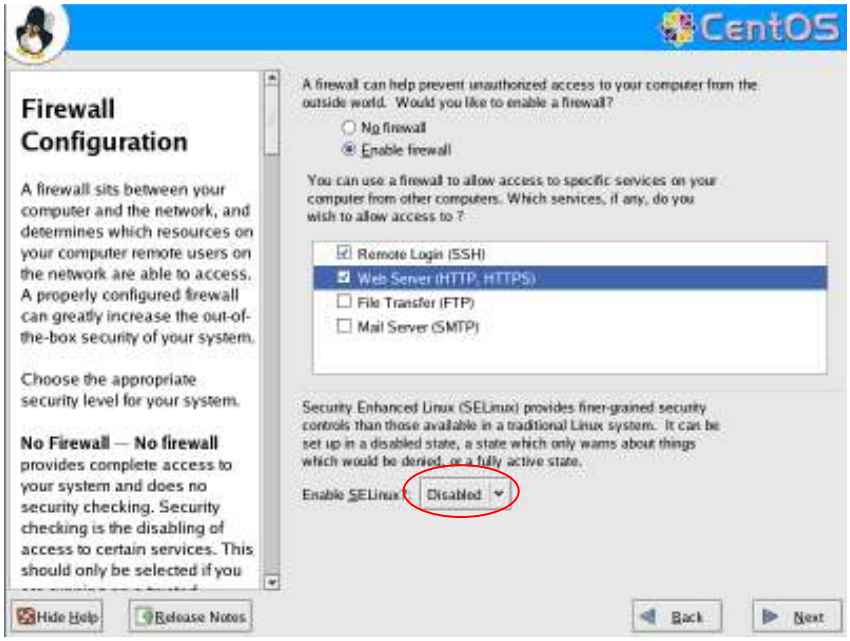


Figura 3-8 Pantalla de Configuración de Cortafuegos [A]

Definición de usuario root: Se debe indicar una contraseña para el usuario root del sistema, como su nombre lo indica, éste será el usuario maestro de la instalación y tendrá control absoluto de acceso sobre toda aplicación en CentOS.

Figura 3-9 Pantalla de Definición Usuario Root [A]

Selección de Aplicaciones o Paquetes: En esta consola tiene la opción de elegir la serie de aplicaciones que se instalarán. Puede agregar aplicaciones a su discreción, tales como un ambiente gráfico u otra función necesaria para cumplir con sus requerimientos.

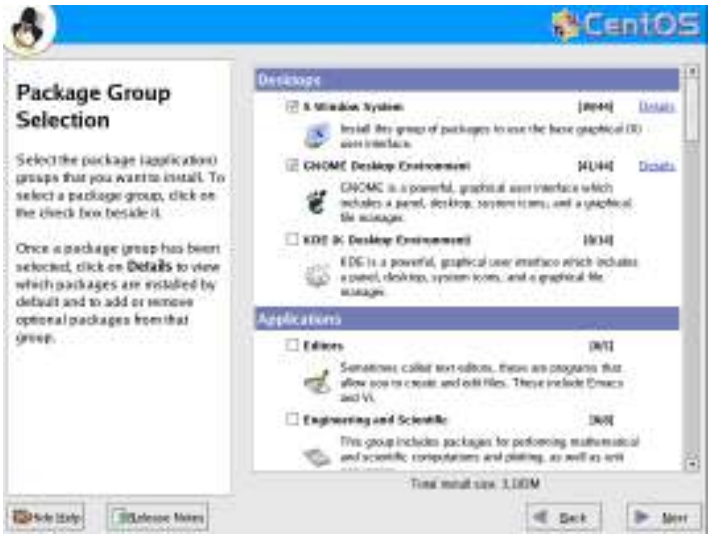


Figura 3-10 Pantalla de Selección de Grupo de Paquetes

Seleccionadas las aplicaciones, al oprimir el botón `Next` iniciará la instalación de aplicaciones, dependiendo del Hardware, este paso puede demorar entre 20 o 40 minutos.

Si la instalación finalizo satisfactoriamente. El sistema arrancara y se mostrará la pantalla para el acceso al mismo:

Figura 3-11 Pantalla de Inicio de CentOS [A]

3.1.2 Configuraciones

3.1.2.1 Configuración IPTABLES

Para crear el firewall con IPTables es necesario crear el archivo que contenga las reglas que este utilizará. El archivo deberá ser ubicado en el directorio /sbin:

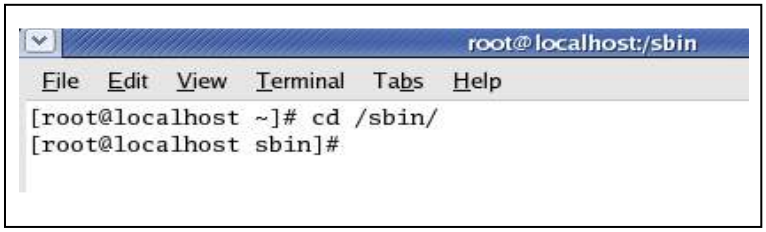


Figura 3-12 Ubicación del Directorio [A]

Se crean el archivo:

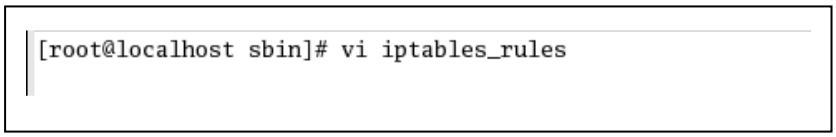


Figura 3-13 Creación del archivo [A]

Se crean las reglas:

```
#!/bin/sh

echo -n Aplicando Reglas de Firewall...

## FLUSH de reglas
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establecemos politica por defecto
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

## Empezamos a filtrar

# El localhost se deja (por ejemplo conexiones locales a mysql)
/sbin/iptables -A INPUT -i lo -j ACCEPT

# A nuestra IP ingresa todo
iptables -A INPUT -s 127.0.0.1 -j ACCEPT
iptables -A INPUT -s 192.168.0.1 -j ACCEPT
```

Figura 3-14 Configuración IPTables 1/2 [A]

```
# El puerto 80 de www debe estar abierto, es un servidor web.
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# Y el resto, lo cerramos
iptables -A INPUT -p tcp --dport 20:21 -j DROP
iptables -A INPUT -p tcp --dport 3306 -j DROP
iptables -A INPUT -p tcp --dport 22 -j DROP
iptables -A INPUT -p tcp --dport 10000 -j DROP

iptables -N allowed
iptables -A allowed -p TCP --syn -j ACCEPT
iptables -A allowed -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A allowed -p TCP -j DROP

iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 192.168.18.128
iptables -A FORWARD -i 192.168.0.1 -j ACCEPT

iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT


echo " OK . Verifique que lo que se aplica con: iptables -L -n"
```

Figura 3-15 Configuración IPTables 2/2 [A]

Se convierte el archivo en ejecutable

```
[root@localhost sbin]# chmod -x iptables_rules
[root@localhost sbin]#
```

Figura 3-16 Archivo ejecutable[A]

Se ejecuta el archivo

```
[root@localhost sbin]# ./iptables_rules
Aplicando Reglas de Firewall... OK . Verifique que lo que se aplica con: iptables -L -n
[root@localhost sbin]#
```

Figura 3-17 Ejecución IPTables[A]

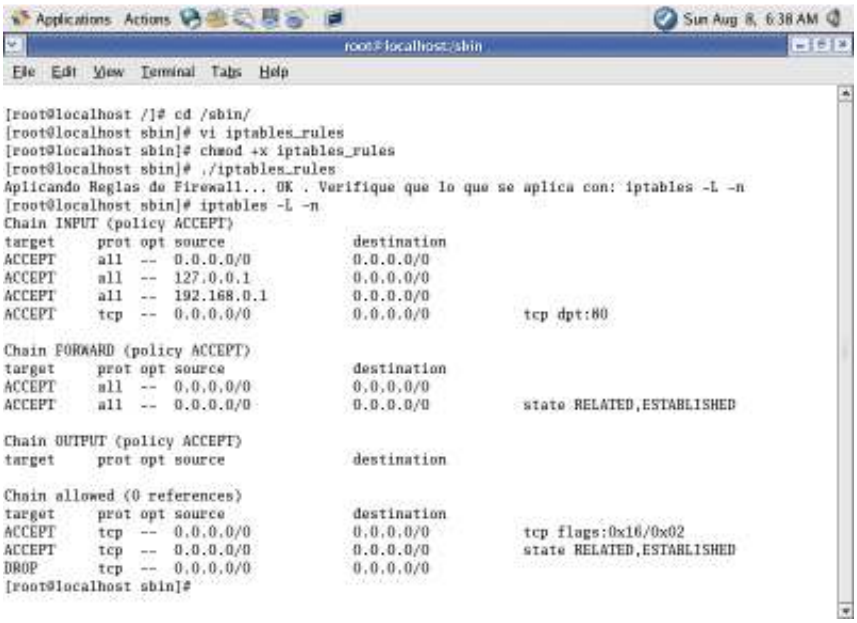


Figura 3-18 Pantalla de Configuración IPTables [A]

Con el comando iptables –L –n podemos verificar lo que aplicamos

```
[root@localhost sbin]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  127.0.0.1            0.0.0.0/0
ACCEPT     all  --  192.168.0.1          0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:80
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpts:20:21
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:3306
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:10000

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain allowed (0 references)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp flags:0x16/0x02
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0
[root@localhost sbin]#
```

Figura 3-19 Verificación de IPTables [A]

3.1.2.2 Configuración CBQ

Para la instalación del CBQ es necesario descargar el archivo cbq.init que se encuentra en el siguiente link: http://sourceforge.net/projects/cbqinit/

Una vez descargado el archivo se coloca en el directorio /sbin.

```
[root@localhost /]cp cbq.init-v0.7.3 /sbin/cbq.init
```

Se convierte el archivo en ejecutable

```
[root@localhost sbin]chmod –x cbq.init
```

```
[root@localhost /]# ls
bin   dev  home    lib         media  mnt  proc  sbin     srv  tmp  var
boot  etc  initrd  lost+found  misc   opt  root  selinux  sys  usr
[root@localhost /]# cd root/
[root@localhost ~]# ls
anaconda-ks.cfg  cbq.init-v0.7.3  Desktop  install.log  install.log.syslog
[root@localhost ~]# cp cbq.init-v0.7.3 /sbin/cbq.init
[root@localhost ~]# cd /sbin/
[root@localhost sbin]# chmod -x cbq.init
[root@localhost sbin]#
```

Figura 3-20 Pantalla de Configuración CBQ.init [A]

Se inicia el servicio del apache:

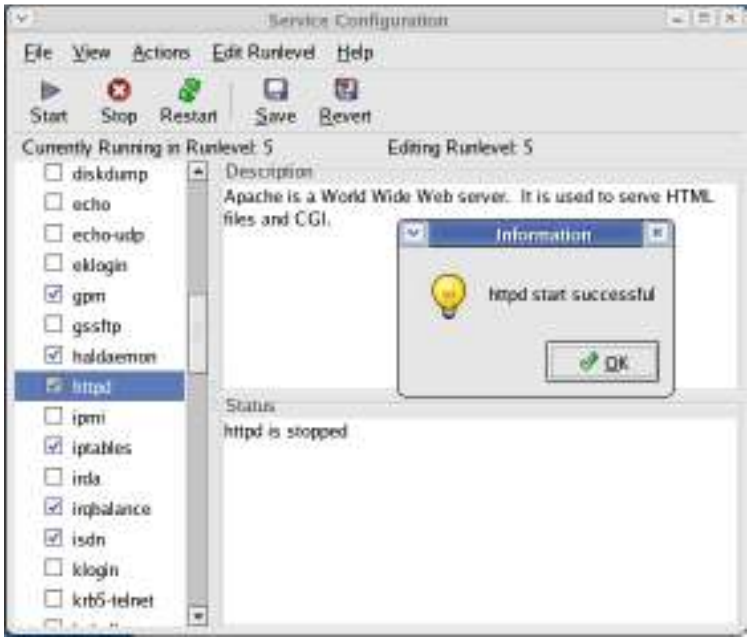


Figura 3-21 Pantalla de Inicio del Servicio Apache [A]

Los archivos de reglas para el CBQ.init deberán ir en el directorio: /etc/sysconfig/cbq.

Se modifican los permisos de los directorios:

```
[root@localhost sbin]# cd /etc/sysconfig/
[root@localhost sysconfig]# chown root:apache /etc/sysconfig/cbq/
[root@localhost sysconfig]# chmod -R 775 /etc/sysconfig/cbq/
[root@localhost sysconfig]#
```

Figura 3-22 Modificación de permisos [A]

Cuando existan archivos en el directorio /etc/sysconfig/cbq podremos ejecutar el siguiente comando:

```
[root@localhost sbin]cbq.init compile
```

```
[root@localhost sbin]# cbq.init compile
/sbin/tc qdisc del dev eth0 root
/sbin/tc qdisc add dev eth0 root handle 1 cbq bandwidth 10Mbit avpkt 1000 cell 8
/sbin/tc class change dev eth0 root cbq weight 1Mbit allot 1514

/sbin/tc qdisc del dev eth1 root
/sbin/tc qdisc add dev eth1 root handle 1 cbq bandwidth 10Mbit avpkt 1000 cell 8
/sbin/tc class change dev eth1 root cbq weight 1Mbit allot 1514

/sbin/tc class add dev eth1 parent 1: classid 1:2 cbq bandwidth 10Mbit rate 10Mbit weight 1Mbit p
rio 8 allot 1514 cell 8 maxburst 20 avpkt 1000

/sbin/tc class add dev eth0 parent 1: classid 1:3 cbq bandwidth 10Mbit rate 10Mbit weight 1Mbit p
rio 8 allot 1514 cell 8 maxburst 20 avpkt 1000

/sbin/tc class add dev eth1 parent 1:0002 classid 1:4 cbq bandwidth 10Mbit rate 128Kbit weight 12
.8Kbit prio 5 allot 1514 cell 8 maxburst 20 avpkt 1000 bounded
/sbin/tc qdisc add dev eth1 parent 1:4 handle 4 tbf rate 128Kbit buffer 10Kb/8 limit 15Kb mtu 150
0
/sbin/tc filter add dev eth1 parent 1:0 protocol ip prio 100 u32 match ip dst 192.168.0.3 classid
 1:4

/sbin/tc class add dev eth1 parent 1:0002 classid 1:5 cbq bandwidth 10Mbit rate 512Kbit weight 51
.2Kbit prio 5 allot 1514 cell 8 maxburst 20 avpkt 1000 bounded
/sbin/tc qdisc add dev eth1 parent 1:5 handle 5 tbf rate 512Kbit buffer 10Kb/8 limit 15Kb mtu 150
0
/sbin/tc filter add dev eth1 parent 1:0 protocol ip prio 100 u32 match ip dst 192.168.0.4 classid
 1:5
```

Figura 3-23 Compilación cbq.init [A]

Una vez compilado podremos iniciarlo:

```
[root@localhost sbin]# cbq.init start
[root@localhost sbin]#
```

Figura 3-24 Inicio cbq.init [A]

Para el CBQ.init podemos utilizar las siguientes opciones:

- **Start.** Inicia el servicio.
- **Compile.** Compila el archivo.
- **Stop.** Para el servicio.
- **Restart.** Reinicia el servicio.
- **List.** Lista los archivos TC.
- **Stats.** Muestra las estadísticas.

3.1.2.3 Configuración MRTG

Para la configuración del MRTG se necesita descargar los paquetes: mrtg, net-snmp-utils, net-snmp

```
[root@localhost /]yum -y install mrtg net-snmp-utils net-snmp
```

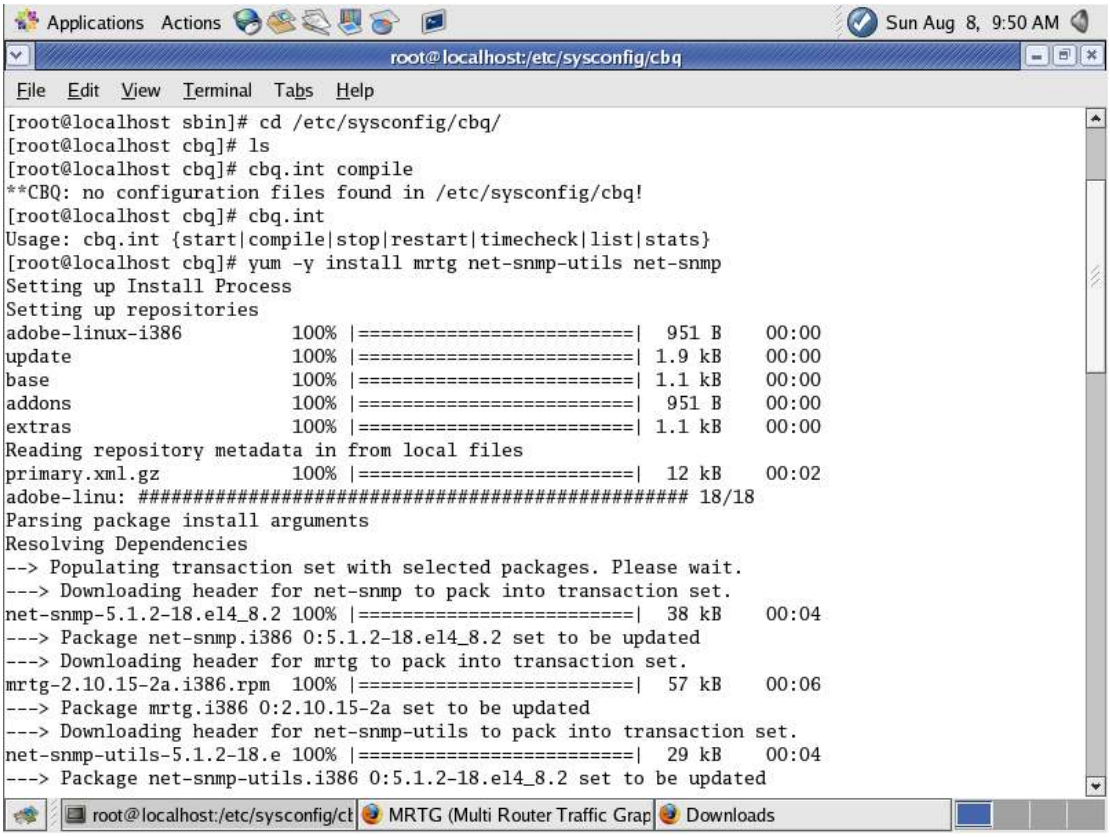


Figura 3-25 Pantalla de Instalación MRTG [A]

Se inicia el servicio de SNMP

```
[root@localhost /]# /etc/init.d/snmpd start
Starting snmpd:                                            [  OK  ]
```

Se configura el archivo snmp.conf

```
# sec.name source community (alias clave de acceso)
com2sec local 127.0.0.1/32 public
com2sec mired 192.168.0.0/24 public
#Se asigna ACL al grupo de lectura escritura
group MyRWGroup v1 local
group MyRWGroup v2c local
group MyRWGroup usm local
#Se asigna ACL al grupo de solo lectura
group MyROGroup v1 mired
group MyROGroup v2c mired
group MyROGroup usm mired
# Ramas MIB que se permiten ver
## name incl/excl subtree mask(optional)
view all included .1 80
# Establece permisos de lectura y escritura
## group context sec.model sec.level prefix read write notif
access MyROGroup "" any noauth exact all none none
access MyRWGroup "" any noauth exact all all all
# Información de Contacto del Scistema
syscontact Administrador (pcpadilla7@hotmail.com)
```

Figura 3-26 Pantalla Archivo SNMP.conf [A]

Se reinicia el servicio SNMP:

```
[root@localhost snmp]# /etc/init.d/snmpd restart
Stopping snmpd:                                            [  OK  ]
Starting snmpd:                                            [  OK  ]
```

Figura 3-27 Pantalla Archivo SNMP.conf [A]

Se crea la carpeta donde se almacenarán los archivos html del MRTG:

```
[root@localhost snmp]# mkdir /home/vhosts/mrtg
```

Figura 3-28 Creación del Directorio MRTG [A]

Se configura el mrtg.conf para el monitoreo de ancho de banda:

```
[root@localhost snmp]# cfgmaker --global "workdir: /home/vhosts/mrtg" -ifref=ip --output /etc/mrt
g/mrtg.cfg --global 'options[_]: growright,bits' public@localhost
```

Figura 3-29 Configuración MRTG [A]

Se crea el archivo índice para el MRTG:

```
[root@localhost mrtg]# indexmaker --output=/home/vhosts/mrtg/index.html /etc/mrtg/mrtg.cfg
```

Figura 3-30 Creación de Índice para MRTG [A]

Se configura el archivo mrtg.conf para que publique las páginas del MRTG:

```
[root@localhost sbin]# vi /etc/httpd/conf.d/mrtg.conf
```

Figura 3-31 Configuración mrtg.conf 1/2 [A]

```
Alias /mrtg /home/vhosts/mrtg

<Location /mrtg>
    Order deny,allow
    Allow from all
    Allow from 127.0.0.1 192.168.0.0/24
</Location>
```

Figura 3-32 Configuración mrtg.conf 2/2 [A]

Se reinicia el servicio de Apache para que lea el nuevo archivo de configuración.

Luego de esta configuración los gráficos pueden ser verificados en el explorador, en la siguiente dirección: http://localhost/mrtg

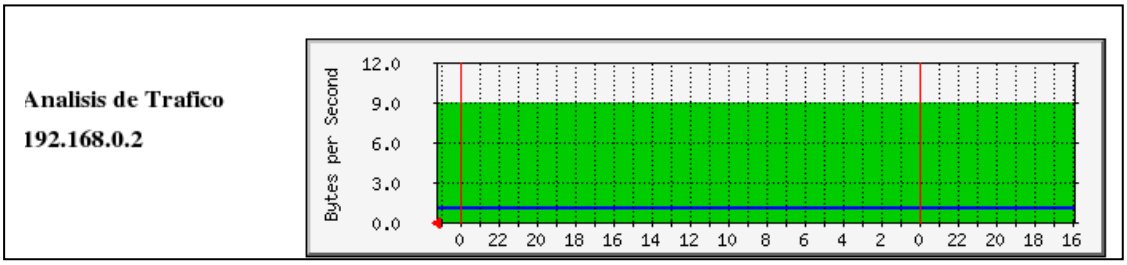


Figura 3-33 Gráficas MRTG [A]

3.1.2.4 Configuración DansGuardian

DansGuardian es un software para el control de contenido, diseñado para controlar el acceso a páginas webs.

El primer paso es descargarlo e instalarlo:

```
[root@localhost sbin]# yum install dansguardian
Setting up Install Process
Setting up repositories
epel                      100% |=========================| 1.1 kB    00:00
dag                       100% |=========================| 1.1 kB    00:00
update                    100% |=========================|  951 B    00:00
rpmforge                  100% |=========================| 1.1 kB    00:00
base                      100% |=========================| 1.1 kB    00:00
Reading repository metadata in from local files
primary.xml.gz            100% |=========================| 2.3 MB    00:37
sqlite cache needs updating, reading in metadata
dag       : ################################################## 11997/11997
primary.xml.gz            100% |=========================| 508 kB    00:06
sqlite cache needs updating, reading in metadata
update    : ################################################## 1179/1179
primary.xml.gz            100% |=========================| 2.3 MB    00:36
sqlite cache needs updating, reading in metadata
rpmforge  : ################################################## 11997/11997
Parsing package install arguments
```

Figura 3-34 Instalación DansGuardian [A]

Una vez instalado, lo configuramos. Es necesario abrir el archivo /etc/dansguardian/dansguardian.conf.

```
vi /etc/dansguardian/dansguardian.conf
```

Eliminamos la línea:

```
UNCONFIGURED - Please remove this line after configuration
```

Si esta línea no estuviera comentada, al reiniciar el aplicativo nos indicaría que el mismo aun no está configurado.

```
# DansGuardian config file for version 2.8.0

# **NOTE** as of version 2.7.5 most of the list files are now in dansguardianf1.conf


# Web Access Denied Reporting (does not affect logging)
#
# -1 = log, but do not block - Stealth mode
#  0 = just say 'Access Denied'
#  1 = report why but not what denied phrase
#  2 = report fully
#  3 = use HTML template file (accessdeniedaddress ignored) - recommended
#
reportinglevel = 3

# Language dir where languages are stored for internationalisation.
# The HTML template within this dir is only used when reportinglevel
# is set to 3. When used, DansGuardian will display the HTML file instead of
# using the perl cgi script.  This option is faster, cleaner
# and easier to customise the access denied page.
# The language file is used no matter what setting however.
#
languagedir = '/etc/dansguardian/languages'

# language to use from languagedir.
language = 'spanish'
```

Figura 3-35 Configuración DansGuardian [A]

El servicio se inicia desde un terminal digitando: dansguardian. Para restringir el contenido de una página Web, abriremos el archivo /etc/dansguardian/bannedsitelist, y en la zona #List other sites to block, se añade la pagina Web que bloquearemos.

```
#As of DansGuardian 2.7.3 you can now include
#.tld so for example you can match .gov for example

#The 'grey' lists override the 'banned' lists.
#The 'exception' lists override the 'banned' lists also.
#The difference is that the 'exception' lists completely switch
#off *all* other filtering for the match.  'grey' lists only
#stop the URL filtering and allow the normal filtering to work.

#An example of grey list use is when in Blanket Block (whitelist)
#mode and you want to allow some sites but still filter as normal
#on their content

#Another example of grey list use is when you ban a site but want
#to allow part of it.

#To include additional files in this list use this example:
#.Include</etc/dansguardian/anotherbannedurllist>

#You can have multiple .Includes.

#List other sites to block:

badboys.com
cholotube.com
xxx.com
facebook.com
```

Figura 3-36 Bloquear páginas en DansGuardian [A]

Con este aplicativo se puede bloquear las descargas de archivos .exe o inclusive bloquear páginas que contengan palabras específicas.

```
#Banned extension list

# File extensions with executable code

# The following file extensions can contain executable code.
# This means they can potentially carry a virus to infect your computer.

.ade  # Microsoft Access project extension
.adp  # Microsoft Access project
.asx  # Windows Media Audio / Video
.bas  # Microsoft Visual Basic class module
.bat  # Batch file
.cab  # Windows setup file
.chm  # Compiled HTML Help file
.cmd  # Microsoft Windows NT Command script
.com  # Microsoft MS-DOS program
.cpl  # Control Panel extension
.crt  # Security certificate
.dll  # Windows system file
.exe  # Program
.hlp  # Help file
.ini  # Windows system file
.hta  # HTML program
.inf  # Setup Information
.ins  # Internet Naming Service
.isp  # Internet Communication settings
# .js   # JScript file - often needed in web pages
```

Figura 3-37 Bloquear extensiones en DansGuardian [A]

Para añadir una palabra a la base, se abre el archivo */etc/dansguardian/bannedphraselist* y se modifica añadiendo la palabra que se necesite.

```
# To block any page containing the words/strings "sex" and "fetish".
# <sex>,<fetish>
#
# < test> will match any word with the string 'test' at the beginning
# <test > will match any word with the string 'test' at the end
# <test> will match any word with the string 'test' at any point in the word
# < test > will match only the word 'test'
# <this is a test phrase> will match that exact phrase
# <test>,<secondtest> will match if both words are found in the page
# A combination of the above can also be used eg < test>,<secondtest>
#
#
# Extra phrase-list files to include
# .Include</etc/dansguardian/testphrase>
#
#
# All phrases need to be within < and > to work, otherwise they will be
# ignored.

# MORE EXAMPLE LISTS CAN BE DOWNLOADED FROM DANSGUARDIAN.ORG

# Phrase Exceptions are no longer listed in this file, they are now
# listed in the exceptionphraselist file.
#
<sex>
.Include</etc/dansguardian/phraselists/pornography/banned>
.Include</etc/dansguardian/phraselists/illegaldrugs/banned>
.Include</etc/dansguardian/phraselists/gambling/banned>
```

Figura 3-38 Bloquear extensiones en DansGuardian [A]

3.1.3 Diseño de la Aplicación

En esta fase diseñaremos el entorno gráfico del sistema. La interfaz front-end ha sido creada de una manera estandarizada y sencilla para que sea amigable con el usuario.

3.1.3.1 Diseño de la Interfaz de Usuario

La aplicación consta de las siguientes interfaces:

- Interfaz Web de Inicio.
- Interfaz Web de Aplicativo.
- Interfaz Web de Análisis de Trafico.
- Interfaz Web de Análisis de Tráfico por PC.

3.1.3.2 Diseño Individual

Interfaz Web de Inicio

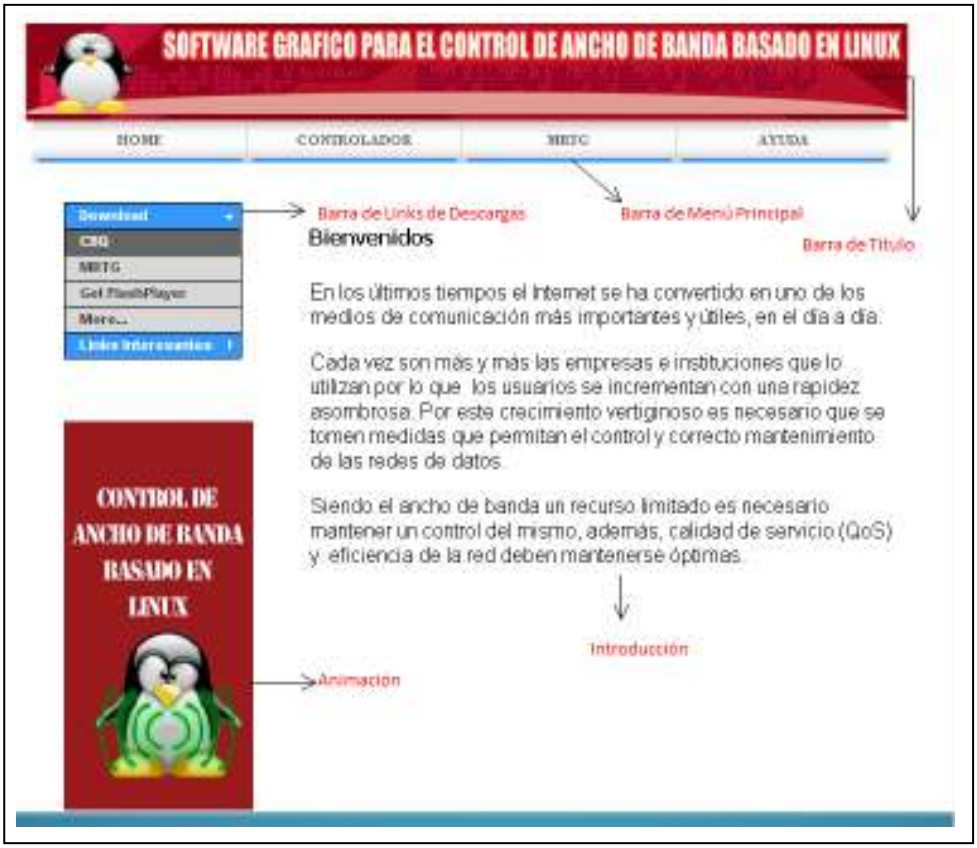


Figura 3-39 Pantalla Principal del Aplicativo [Á]

Esta interfaz está compuesta por:

- Barra de Titulo
- Barra de Menú Principal
- Barra de Links de Descarga
- Introducción
- Animación

Barra de Titulo.

Figura 3-40 Barra de Título [A]

En esta barra tenemos el titulo de la aplicación.

Barra de Menú Principal.

Figura 3-41 Barra de Menú Principal [A]

Esta barra está compuesta por varios botones los cuales son:

Home.- Ingresa a la página principal.

Controlador.- Ingresa a la aplicación como tal.

MRTG.- ingresa al índice de los gráficos de MRTG.

Acerca De.- Muestra una breve descripción de la licencia de la aplicación.

Contáctenos.- Información de correo electrónico de la desarrolladora.

Barra de Links de Descarga.

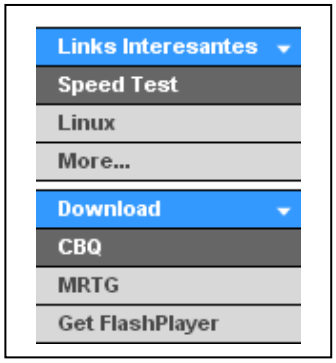


Figura 3-42 Barra de Links de Descarga [A]

En está barra se han colocado varios links para descargar los fuentes de las aplicaciones usadas en esta tesis, como son: CBQ, MRTG y Flash Player. También se han colocado links interesantes como Speed Test, que ayuda a verificar la velocidad de nuestro internet y una página para estar informado sobre las noticias de Linux.

Interfaz Web de Aplicativo.

En esta interfaz podremos verificar los padres que existen (interfaces de red), las reglas creadas y modificar a cualquiera de ellas. Verificar el la regla que se crea.

Además existe una barra de estado para el archivo CBQ.init, desde donde se puede iniciar, parar, reiniciar y compilar el mismo.

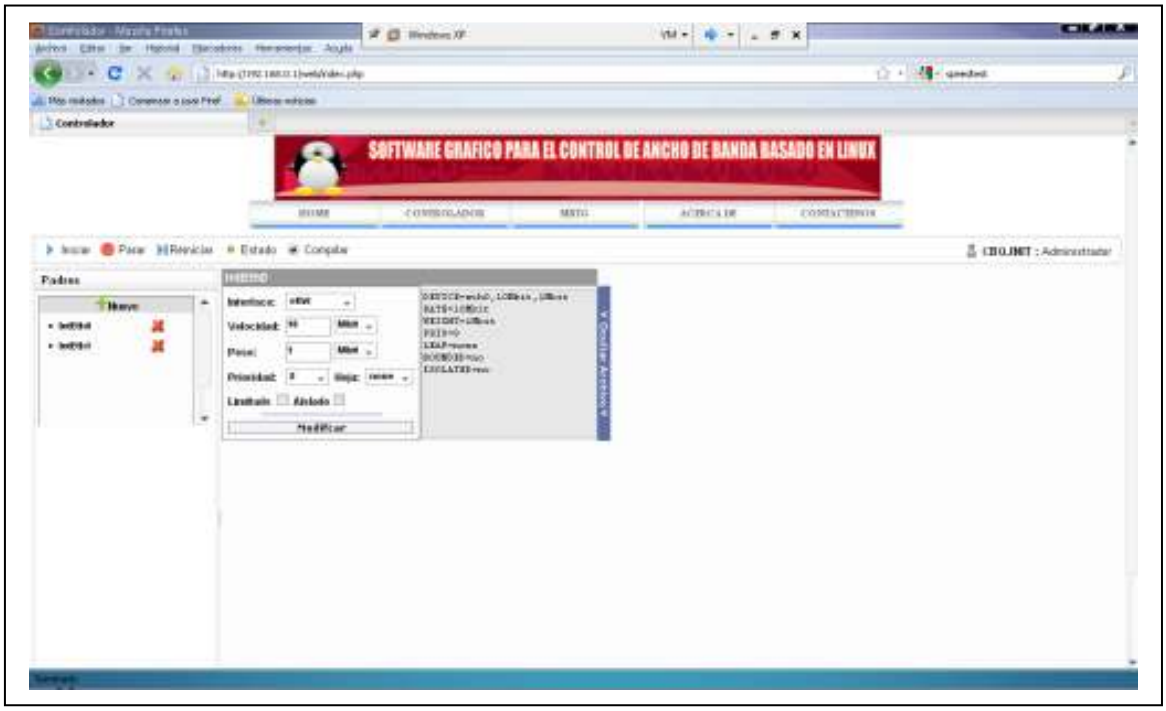


Figura 3-43 Pantalla de Interfaz Web de Aplicativo [A]

Esta interfaz está compuesta de:

- Barra de Titulo
- Barra de Menú
- Barra de Estado de CBQ.init
- Listado de Padres
- Listado de Reglas
- Área de Modificación de Regla
- Área para mostrar archivo

Interfaz Web de Análisis de Tráfico

Esta interfaz mostrará las maquinas que se encuentran en nuestra red y que su ancho de banda está siendo verificado por la aplicación MRTG.

Cada uno de las graficas presentadas en está pagina tienen un link que abrirá una página de análisis de trafico de esa PC.

Figura 3-44 Pantalla de Interfaz Web de Análisis de Tráfico [A]

Web Análisis de Tráfico por PC.

Cada PC que se esté verificando con el aplicativo tendrá una página que contiene de información anual, mensual y semanal del ancho de banda que utilizó.

Figura 3-45 Pantalla de Gráficas MRTG [A]

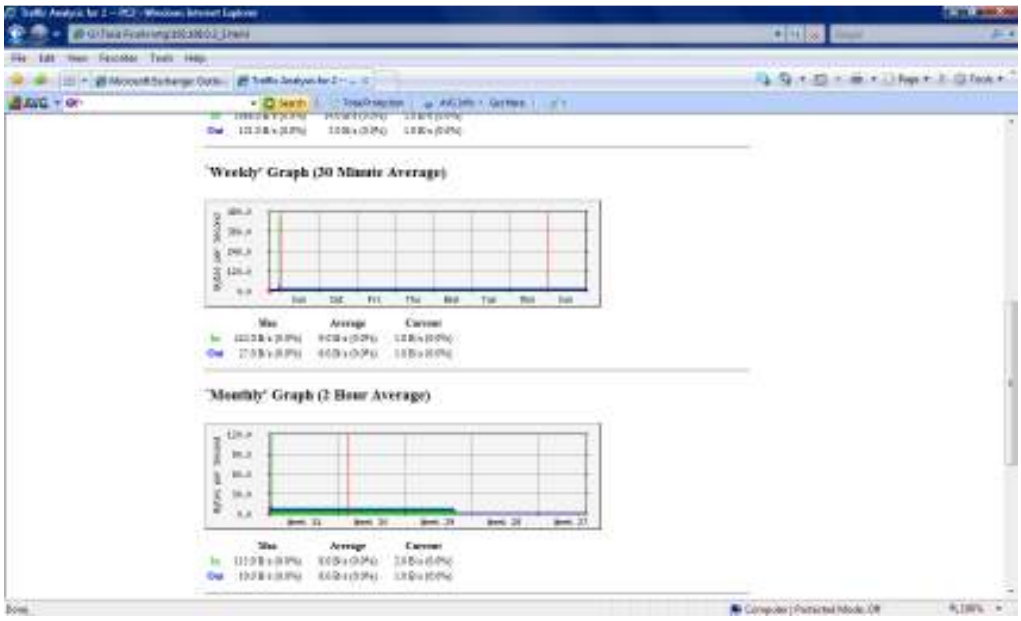


Figura 3-46 Pantalla de Gráficas MRTG [A]

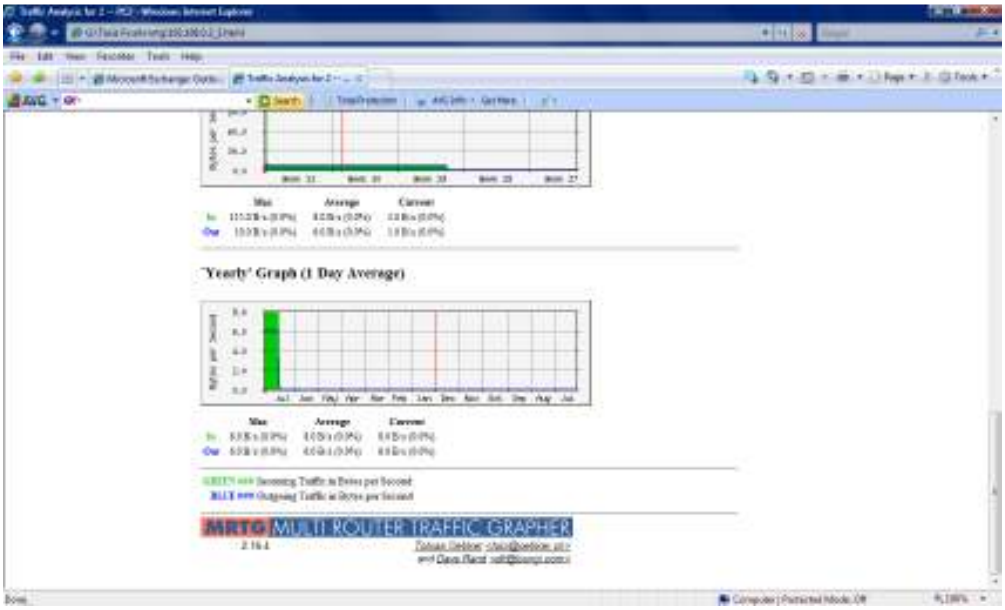


Figura 3-47 Pantalla de Gráficas MRTG [A]

CAPITULO IV

4 PRUEBAS EN UNA RED ETHERNET

4.1 Velocidad de Transferencia

Antes de realizar las pruebas es necesario explicar que es y cómo se mide la velocidad de transferencia de datos.

Al momento de crear las reglas existe un dato que es muy importante y esencial, el parámetro RATE, este nos indicara la velocidad de ancho de banda que se asigna a un equipo de la red. Esta velocidad es denominada velocidad binaria (bit rate), y de acuerdo al Sistema Internacional de Unidades, la unidad con la que se expresa esta velocidad es el bit por segundo, es decir bit/s, b/s o bps, donde la b siempre debe escribirse en minúscula para impedir confusión con la unidad byte por segundo (B/s). Los múltiplos para byte aplican de diferente modo que para bit. La unidad byte es igual a 8 bits, y a partir de esto se puede utilizar la siguiente tabla:

UNIDADES	DESCRIPCION
kbit/s o kbps (kb/s, kilobit/s) Mbit/s o Mbps(Mb/s, Megabit/s)	1000 bits por segundo 1 millón de bits por segundo
Gbit/s o Gbps (Gb/s, Gigabit/s)	Mil millones de bits por segundo
byte/s (B/s)	8 bits por segundo
kilobyte/s (kB/s, mil bytes)	8 mil bits por segundo
megabyte/s (MB/s, un millón de bytes)	8 millones de bits por segundo
gigabyte/s (GB/s, mil millones de bytes)	8 mil millones de bits

Cuadro 4-1 Tabla de Unidades de Velocidad [A]

4.2 Parámetros para las Reglas CBQ

Antes de iniciar la configuración, se deben determinar los valores para los siguientes parámetros. Para construir una regla, se requiere al menos comprender y especificar los valores para los parámetros DEVICE, WEIGHT, RATE y RULE. Las reglas pueden ser tan complejas como la imaginación del administrador lo permita.

Los ficheros con las configuraciones se guardan dentro del directorio /etc/sysconfig/cbq/ y deben llevar la siguiente nomenclatura:

```
cbq-[número-ID-Clase].[nombre]
```

Donde número-ID-Clase corresponde a un número hexadecimal de 2 bits dentro del rango 0002-FFFF. Ejemplo:

```
cbq-0002.smtp-in
```

Parámetro DEVICE.

Es un parámetro obligatorio. Se determina los valores con el nombre de la interfaz, ancho de banda y peso de esta interfaz. Este último valor, que es opcional en este parámetro, se calcula dividiendo el ancho de banda de la interfaz entre diez. Por ejemplo, sí se dispone de una interfaz denominada eth0 de 100 Mbit/s, el peso será 10 Mbit/s, de tal modo los valores del parámetro DEVICE.

Parámetro de clase RATE.

Es un parámetro obligatorio. Se refiere al ancho de banda a asignar a la clase. El tráfico que pase a través de esta clase será modificado para ajustarse a la proporción definida. Por ejemplo, sí se quiere limitar el ancho de banda utilizado a 10 Mbit/s, el valor de RATE sería 10Mbit.

Parámetro de clase WEIGHT.

Es un parámetro obligatorio. Éste es proporcional al ancho de banda total de la interfaz. Como regla se calcula dividiendo entre diez el ancho de banda total. Para una interfaz de 2048 kbps, correspondería un valor de 204Kbit:

Parámetro de clase PRIO.

Es un parámetro opcional que se utiliza para especificar que prioridad tendrá sobre otras reglas de control de ancho de banda.

Mientras más alto sea el valor, menos prioridad tendrá sobre otras reglas. Se recomienda utilizar el valor 5 que funcionará para la mayoría de los casos.

Parámetro de clase BOUNDED.

Es un parámetro opcional. Si el valor es yes, que es el valor predeterminado, la clase no tendrá permitido utilizar ancho de banda de la clase padre.

Si el valor es no, la clase podrá hacer uso del ancho de banda disponible en la clase padre. Si se establece con valor no, es necesario utilizar none o bien sfq en el parámetro LEAF.

Parámetro de clase ISOLETED.

Es un parámetro opcional. Si se establece con el valor yes, la clase no prestará ancho de banda a las clases hijas. Si se utiliza el valor no, que es el valor predeterminado, se permitirá prestar el ancho de banda disponible a las clases hijas.

Parámetros de filtración.

Son las reglas de filtración que se utilizan para seleccionar tráfico en cada una de las clases. La sintaxis completa es la siguiente:

```
RULE=[[saddr[/prefijo]][:puerto[/máscara]],][daddr[/prefijo]][:puert
o[/máscara]]
```

Donde saddr se refiere a la dirección de origen. Daddr a la dirección de destino.

La sintaxis simplificada es la siguiente, donde todos los valores son opcionales, pero se debe especificar al menos uno:

```
RULE=IP-origen:puerto-origen,IP-destino:puerto-destino
```

En general la interpretación sigue cuatro simples principios:

Cualquier dirección IP o red que se coloque antes de la coma se considera dirección IP o red de origen.

Cualquier dirección IP o red que se coloque después de la coma se considera dirección IP o red de destino.

Cualquier puerto antes de la coma se considera el puerto de origen.

Cualquier puerto especificado después de la coma se considera puerto de destino.

4.3 Pruebas de Controlador de Ancho de Banda

Para realizar las pruebas se creó una pequeña red Ethernet que está formada por 3 equipos con Windows XP y 1 equipo con sistema operativo CentOS 4.7 con kernel 2.6.9 al cual lo denominaremos como el servidor.

Las IP's del servidor Linux son:

IP ETH0: 192.168.18.128 (Asignada por DHCP)

Mascara de Red: 255.255.255.0

Brodcast: 192.168.18.255

IP ETH1: 192.168.0.1 (Red LAN)

Mascara de Red: 255.255.255.0

Brodcast: 192.168.0.255

Las IP's de las pc's con Windows son:

PC1:

Dirección IP: 192.168.0.2

Mascara de Red: 255.255.255.0

Puerta de Salida: 192.168.0.1

PC2:

Dirección IP: 192.168.0.3

Mascara de Red: 255.255.255.0

Puerta de Salida: 192.168.0.1

PC3:

Dirección IP: 192.168.0.4

Mascara de Red: 255.255.255.0

Puerta de Salida: 192.168.0.1

Las tres máquinas utilizan al servidor Linux como proxy para salir a internet. Las pruebas que realizaremos son las siguientes:

PC1, Velocidad 64Kbit

PC2, Velocidad 128Kbit

PC3, Velocidad 256Kbit

Intentaremos bajar un mismo archivo en los 3 equipos al tiempo para verificar que ocurre. Al momento se tiene 2 interfaces de red, por esta razón se crean 2 padres, uno por cada interfaz de red. El primer paso es crear las reglas:

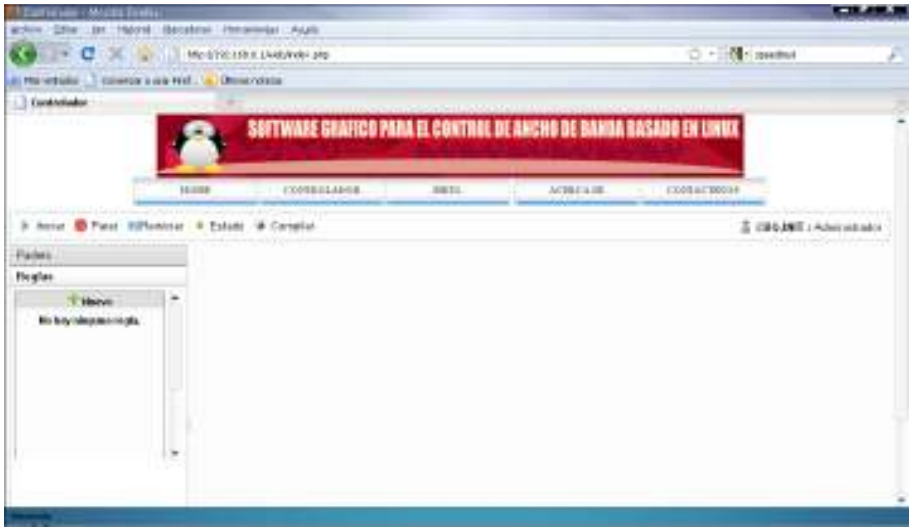


Figura 4-1 Pantalla Interfaz Gráfica [A]

Para las reglas las crearemos como: PC1, PC2, PC3.

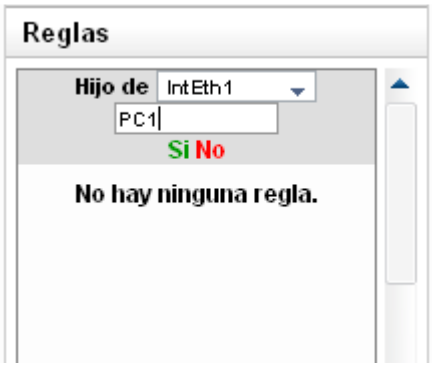


Figura 4-2 Pantalla Creación de Reglas [A]

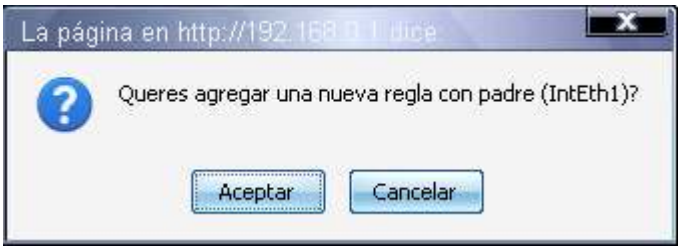


Figura 4-3 Cuadro de Dialogo para creación de Reglas [A]

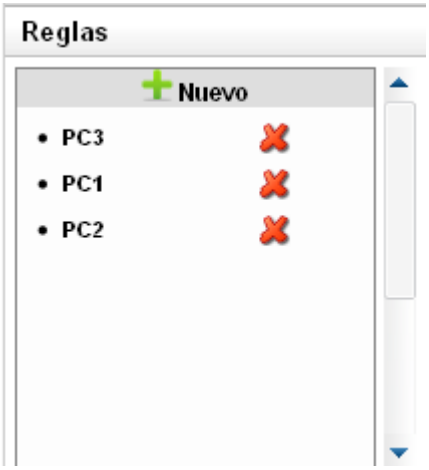


Figura 4-4 Pantalla Reglas Creadas [A]

Luego de haberlas creado, modificaremos los archivos para colocar las velocidades indicadas en la página anterior.

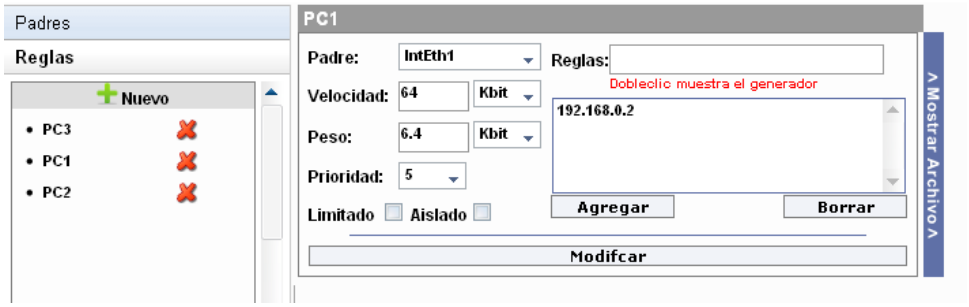


Figura 4-5 Pantalla Creación de Reglas [A]

Figura 4-6 Pantalla Creación Reglas [A]

Figura 4-7 Pantalla Creación Reglas [A]

Compilamos e iniciamos el servicio de CBQ.init.

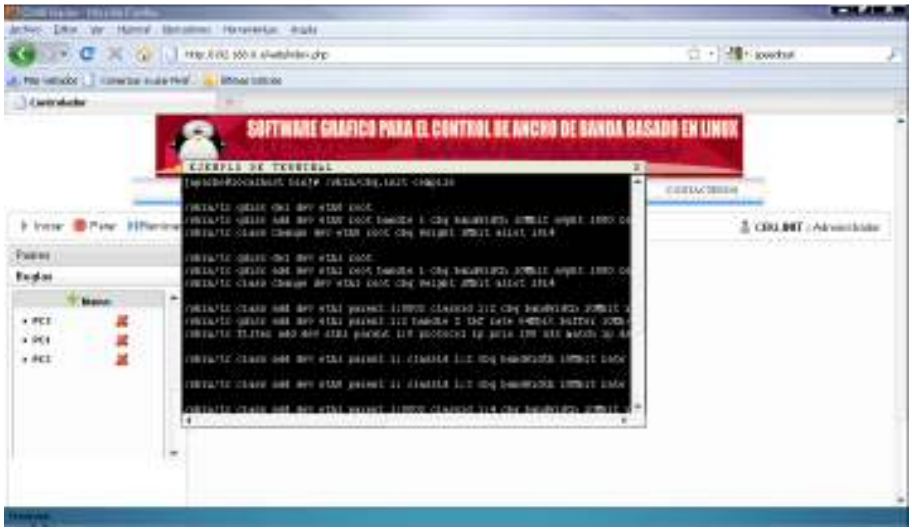


Figura 4-8 Pantalla Compilación CBQ.init [A]

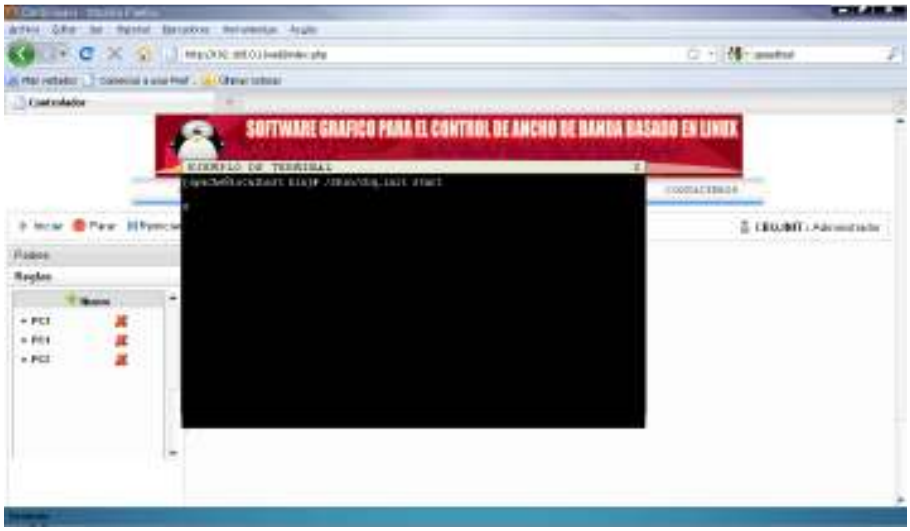


Figura 4-9 Pantalla Inicio de Servicio CBQ.init [A]

El archivo esta compilado e iniciado, ahora si intentamos bajar un archivo, por ejemplo: la imagen de un disco de CentOS, lo bajaremos de su página principal.

Figura 4-10 Pantalla Página de Descarga [A]

Al mismo tiempo de empezada la descarga tenemos los datos:

PC1

Tasa de Transferencia: 4.16KB/s

Megas Descargados: 578KB

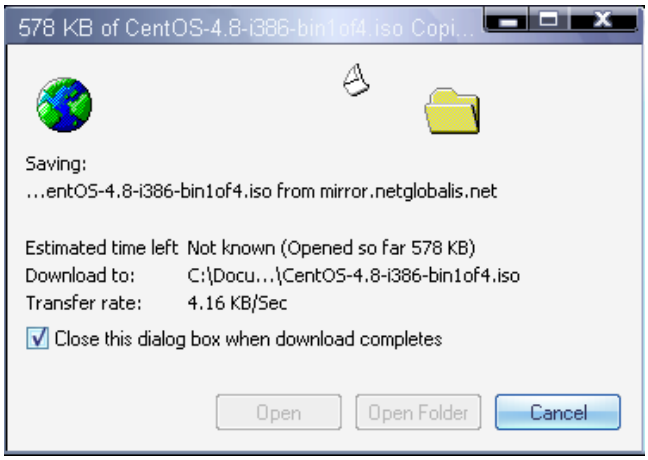


Figura 4-11 Pantalla Descarga PC 1 [A]

Figura 4-12 Pantalla Velocidad de PC 1 [A]

PC2

Tasa de Transferencia: 5.5KB/s

Megas Descargados: 1.2MB

Figura 4-13 Pantalla Descarga PC 2 [A]

Figura 4-14 Pantalla Velocidad PC 2 [A]

PC3

Tasa de Transferencia: 13.5KB/s

Megas Descargados: 580KB

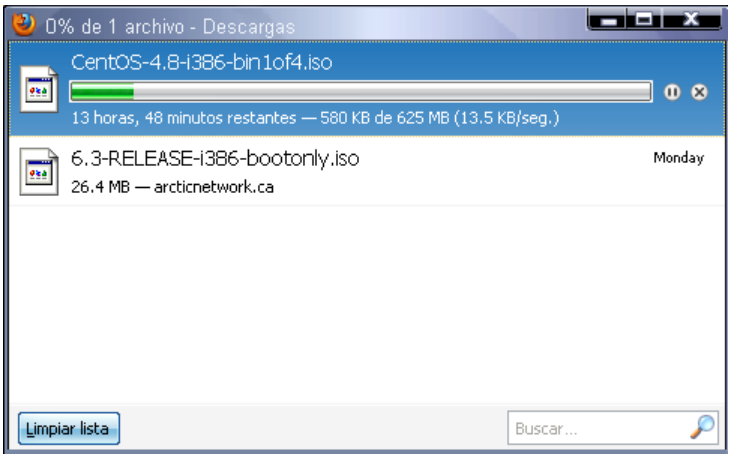


Figura 4-15 Pantalla Descarga PC 3 [A]

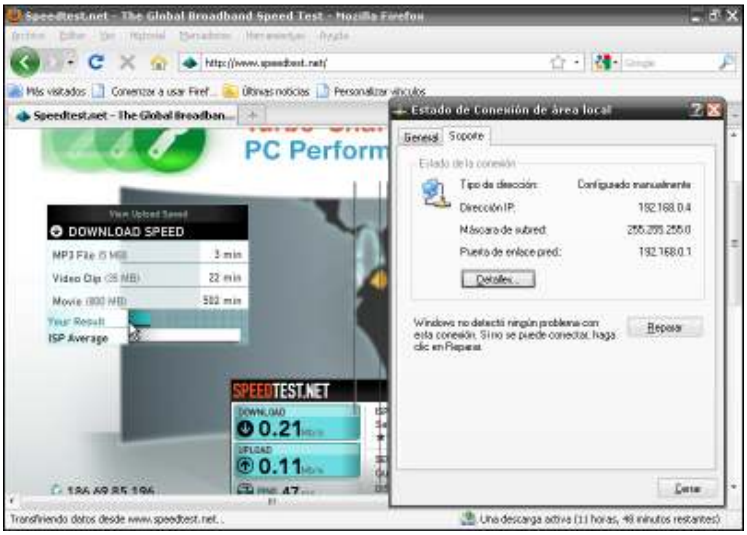


Figura 4-16 Pantalla Velocidad PC 3 [A]

En cada una de las pantallas capturadas se puede observar que se verifico la velocidad del ancho de banda y al momento cada una de ellas muestra el valor que se encuentra en las reglas.

La prueba realizada, es una prueba sencilla en la cual podemos observar que el CBQ está funcionando sin problemas.

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

El software con una interfaz gráfica basada en Web facilita el control de ancho de banda y seguridades, ya que en cualquier momento el administrador puede ingresar desde una máquina en la red a la interfaz gráfica para realizar el monitoreo y control de la red.

GNU/Linux en sus diferentes distribuciones, posee herramientas muy potentes para el control de ancho de banda y seguridades, aunque poco amigables. Las herramientas utilizadas en la aplicación se basan en comandos y configuraciones realizadas en modo texto, las mismas fueron: IPTables, Squid, DansGuardian y CBQ.

Debido a que CBQ es una herramienta muy flexible y fácil de utilizar, fue seleccionada como base para realizar el desarrollo de la interfaz web.

Las soluciones existentes al momento en el mercado para Firewalls, tienen costos muy elevados por lo que no son accesibles para una empresa que posee una red mediana o pequeña. Con un servidor Linux se puede crear soluciones muy potentes de Firewall, con un costo muy bajo ya que las distribuciones se pueden descargar del internet.

La interfaz gráfica creada es muy amigable con el usuario, por esta razón no es necesario que el administrador sea un experto en GNU/Linux, sin embargo es preciso que conozca de redes y como el aplicativo realiza el control de ancho de banda.

GNU/Linux posee muchas herramientas para el monitoreo de una red como son: MRTG, NTOP, CACTI, IPTraf, entre las más conocidas. Todas estas son de fácil instalación y configuración. MRTG es la herramienta que se utilizó como complemento para el monitoreo en el aplicativo diseñado.

Ya que las herramientas utilizadas están basadas en GNU/Linux, la herramienta debe ser considerada como “Software Libre” con una licencia GPL, es decir que el aplicativo puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

5.2 Recomendaciones

Es fundamental que en la red que se vaya a implementar la aplicación, el administrador deba colocar sus parámetros y ajustarlos a sus necesidades.

Además de MRTG se puede añadir otra herramienta al aplicativo para realizar el monitoreo.

Se recomienda que el aplicativo realizado sea utilizado en pequeñas o medianas empresas, debido a que es una solución para Firewall de bajo costo muy poderosa.

Ya que el aplicativo tiene una licencia GPL es necesario que el administrador estudie el código para que pueda entender cómo funciona el aplicativo y si desea, mejorar el mismo.

El aplicativo podría ser mejorado en una segunda versión incrementado la opción gráfica para administrar DansGuardian, que es utilizado para filtrar contenido en sitios web. Esta es una herramienta poderosa pero administrable solo en modo texto.

Inscribirlo como herramienta de software libre con licencia (GPL) y generar una comunidad para que lo mejoren.

CAPÍTULO VI

6. REFERENCIAS Y BIBLIOGRAFIA

[A] Diseño e Implementación de Software Gráfico para la Administración y Control de Ancho de Banda y Seguridades en Redes basado en Linux, Paola Cristina Padilla Albán, Noviembre 2010, Pontificia Universidad Católica del Ecuador.

[1] http://www.monografias.com/trabajos53/topologias-red/topologias-red.shtml

[2] http://www.monografias.com/trabajos53/topologias-red/topologias-red.shtml

[3] http://www.monografias.com/trabajos53/topologias-red/topologias-red.shtml

[4] http://www.monografias.com/trabajos53/topologias-red/topologias-red.shtml

[5] http://www.monografias.com/trabajos53/topologias-red/topologias-red.shtml

[6] http://www.monografias.com/trabajos53/topologias-red/topologias-red.shtml

[7] http://systemsview.net/compsystems/network/images/osi.jpg

[8]http://koalasoft.homelinux.net/Manuales/wiki/index.php/Cortafuegos_%28inform%E1tica%29

[9] *https://www.tlm.unavarra.es/~eduardo/.../20010403-alcom-espanol.pdf*

http://www.osmosislatina.com/centos/instalacion.htm

http://crazytoon.com/2007/05/01/mrtg-multi-router-traffic-grapher-how-to-setup-mrtg-in-linux-to-monitor-bandwidth-usage/#ixzz0w3qDMIOr