IMPLEMENTACIÓN BÁSICA DE UN SISTEMA PARA EL FILTRADO DE PAQUETES EN UN ENLACE WAN

ANDRÉS AUGUSTO JÁCOME LOBO

TATIANA INÉS NAVAS GÓMEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FISICOMECÁNICAS

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

BUCARAMANGA

2005

IMPLEMENTACIÓN BÁSICA DE UN SISTEMA PARA EL FILTRADO DE PAQUETES EN UN ENLACE WAN

ANDRÉS AUGUSTO JÁCOME LOBO

TATIANA INÉS NAVAS GÓMEZ

Este proyecto es presentado como requisito para optar al título de Ingeniero Electrónico

Director

OSCAR GUALDRÓN GONZÁLEZ, PhD

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FISICOMECÁNICAS

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

BUCARAMANGA

2005

A Dios, por darme vida y por permitirme ser todo lo que soy.

A mi Papi Jorge por darme su apoyo incondicional y por estar siempre a mi lado.

A mi Mami María Inés por acompañarme en todo momento y darme su apoyo todos los días a toda hora.

A mi hermana Kelly por ser mi mejor amiga.

A Sergio por darme su apoyo incondicional y por su cariño.

A Andrés por ser mi compañero en las buenas y en las malas, y que gracias a el pudimos sacar el proyecto adelante.

A toda mi familia y amigos que de alguna u otra manera me ayudaron para lograr esta meta, por haber permitido que sacrificara tiempo valioso de su compañía para poder terminar mi carrera.

Tatiana Inés Navas Gómez

A Dios, por darme vida y felicidad, por todas las veces que me inspiró e iluminó para poder sacar adelante este proyecto y por permitirme ser todo lo que soy.

A mi Papá Gabriel por apoyarme incondicional mente y por darme fuerzas para seguir siempre adelante.

A mi Mamá Martha por acompañarme y apoyarme en todo momento y por darme motivos para salir adelante.

A mis hermanas y hermanos por esperar siempre de mí lo mejor.

A Lorena por darme su apoyo incondicional, por quererme y por mantenerme siempre alegre, animado y fortalecido para poder trabajar bien.

A mi mejor amiga Tatiana por ser una excelente compañera de trabajo.

Y a todos los familiares que me ayudaron a lo largo de la carrera, porque sin ellos no habría podido conquistar esta meta.

Andrés Augusto Jácome Lobo

AGRADECIMIENTOS

Quienes elaboraron este proyecto, agradecen a todas aquellas personas que nos sirvieron de apoyo y fueron de gran ayuda en el desarrollo del proyecto, pero muy especialmente a nuestro director Oscar Gualdrón González por creer en nosotros y por apoyarnos siempre; al Ingeniero Jaime Rueda Rivera porque gracias a su apoyo y a su gran ayuda fue posible terminar a tiempo nuestra labor; al Ingeniero Samuel Gonzalo Pinzón, a la Ingeniera Leidy Barco y a José Miguel Aguilera por la orientación prestada y por su oportuna colaboración.

A nuestros familiares y amigos, en quienes encontramos un soporte constante durante la ejecución de este proyecto.

A la Especialización en Telecomunicaciones, a la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones y a la Universidad Industrial de Santander.

TABLA DE CONTENIDO

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

LISTA DE ANEXOS

TITULO IMPLEMENTACIÓN BÁSICA DE UN SISTEMA PARA EL FILTRADO DE PAQUETES EN UN ENLACE WAN [*]

ANDRÉS AUGUSTO JÁCOME LOBO
TATIANA INÉS NAVAS GÓMEZ [**]

Palabras claves: *Filtrado de paquetes, Redes de área extensa (WAN), Reglas de filtrado, Adaptadores WAN SyncLink para Linux, Linux como Router, Firewall en Linux, Rendimiento, Políticas de Seguridad.*

En este trabajo se presenta la implementación de un sistema para el filtrado de paquetes en un enlace WAN con el fin de tratar de mejorar la transmisión de datos a través de largas distancias, proporcionando mayor seguridad y un mejor rendimiento del enlace, además se encuentran los conceptos básicos acerca de la configuración de redes WAN filtrado de paquetes haciendo énfasis en las reglas de filtrado, se describe brevemente el mecanismo desarrollado por los autores para realizar el filtrado de paquetes en un enlace WAN contando con los adaptadores SyncLink de Microgate ®, y finalmente se muestran las pruebas realizadas para comprobar el desempeño del enlace en condiciones de operación real contando con el software MGEN y DREC los cuales se encargan de la generación y recepción de paquetes con o sin tráfico respectivamente.

Para cumplir con este fin se propone un estudio que contempla la configuración e instalación de un enlace WAN entre dos estaciones de trabajo ubicadas en el laboratorio de redes de datos de la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones de la Universidad Industrial de Santander; y con el fin de generar posibles herramientas que permitan ejecutar un mecanismo para realizar filtrado de paquetes según reglas y requerimientos previamente establecidos.

---

* Trabajo de Grado

** Facultad de Ingenierías Fisicomecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Oscar Gualdrón González, PhD.

TITLE BASIC IMPLEMENTATION OF A SYSTEM FOR THE FILTRATE OF PACKAGES IN A WAN LINK *

ANDRÉS AUGUSTO JÁCOME LOBO
TATIANA INÉS NAVAS GÓMEZ **

Key words: *Filtrate of packages, Wide Area Network (WAN), Rules of filtrate, Adapters WAN SyncLink for Linux, Linux working like Router, Firewall in Linux, Yield, Policies of Security.*

In this work the implementation of a system for the filtrate of packages in a WAN link is shown in order to improve the data transmission through long distances, by providing greater security and a better yield of the link, where the basic concepts about the configuration of Wide Area Networks take place, as well as packets filtering with emphasis in the filtrate rules. In addition to this, it is shortly described the mechanism developed by the authors to make the packets filtering in a WAN link being counted with the Microgate ® SyncLink adapters. Finally in this document there are the tests that have been made to verify the performance of the WAN link in conditions of real operation using different kind of software like MGEN and DREC which have the special function of the generation and reception of packages with or without traffic.

In order to fulfill this aim it is necessary to set out a study that contemplates the configuration and installation of a WAN link between two located workstations in the laboratory of data networks of the School of Electrical, Electronic and Telecommunications Engineering of the Industrial University of Santander; and with the purpose of generating possible tools that allow to execute a mechanism to make filtrate of packages according to rules and requirements previously established

---

* Work for Graduation

** Faculty of physic-mechanical Engineerings , School of Electrical, Electronic and Telecommunications Engineerings. Director: Oscar Gualdrón González, PhD.

INTRODUCCIÓN

El estudio de las redes de datos ha abarcado especialmente los campos relacionados con las redes más comunes y cercanas al hombre como es el caso de las redes de área local. Sin embargo en la actualidad los avances en la industria tecnológica han permitido que la transmisión de datos alcance un nuevo nivel permitiendo al hombre utilizar sistemas de comunicación cada vez más rápidos, más confiables y con mayor cobertura. Así mismo las comunicaciones a larga distancia cada día son más utilizadas con el fin de acortar distancias y disminuir el tiempo de acceso a la información deseada.

En este orden de ideas, y con el fin de buscar soluciones a problemas en la comunicación entre redes a larga distancia se propone la implementación de un sistema para el filtrado de paquetes dentro de un enlace WAN que permite mejorar su rendimiento al evitar el transporte de paquetes no deseados entre redes.

Para cumplir con este fin se propone un estudio que contempla la configuración e instalación de un enlace WAN entre dos estaciones de trabajo ubicadas en el laboratorio de redes de datos de la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones y con base en este estudio generar posibles herramientas que permitan ejecutar un mecanismo para realizar el filtrado de paquetes según reglas y requerimientos previamente establecidos.

Finalmente se busca analizar el desempeño del enlace dentro de un ambiente con condiciones controladas las cuales representan las características propias del medio donde opera el enlace y de acuerdo a la respuesta encontrada determinar el rendimiento del enlace en situaciones de operación real.

1. CONCEPTOS FUNDAMENTALES DEL FILTRADO DE PAQUETES EN UNA RED DE AREA EXTENSA.

Antes de entrar en el proceso de filtrado de paquetes en una WAN[1], es necesario puntualizar algunos conceptos básicos acerca de redes de datos.

1.1. CONCEPTOS GENERALES DE UNA RED DE DATOS

Para empezar, una red de datos consiste en un conjunto de elementos que tienen como fin común compartir recursos. Entre sus principales objetivos se encuentra hacer que todos los programas, datos, y equipos estén disponibles para cualquier dispositivo de la red que así lo solicite, sin importar la localización física del recurso o del usuario.

Esta red proporciona una alta fiabilidad, seguridad, y rapidez en cada una de las operaciones que se realiza entre estaciones. Estas características se logran gracias a que dentro de cada red se crean las condiciones necesarias para la interconexión de puntos de trabajo y el manejo conjunto de recursos. Como resultado, en la red se obtiene un mejor rendimiento y una mayor seguridad y fiabilidad de los datos que son transmitidos, dado que se conoce con que tipo de estaciones se esta trabajando.

Ahora, como complemento a las características de una red de datos presentadas anteriormente, es necesario decir que una WAN se especializa en la interconexión de redes. Esta unión mantiene todas las propiedades de cada una de las redes independientes y agrega la posibilidad de realizar operaciones

[1] Wide Area Network, Redes de área extensa. Redes de gran cobertura

de larga distancia con poco retardo y bajo costo.

Para la conformación de una WAN se tienen ciertos requerimientos básicos que permiten lograr un nivel de desempeño aceptable, entre los cuáles están:

- Conformación previa de redes de área local (LAN) en cada extremo del enlace. Estas redes están encargadas de dar soporte a la conexión prestando sus elementos básicos como nuevos integrantes en el enlace WAN.

Figura 1. Esquema general de una WAN

- Contar con líneas de transmisión en perfecto estado y listas para su funcionamiento inmediato. Estas líneas de transmisión tienen como función mover los bits desde una estación transmisora a una estación receptora.

- Contener elementos de conmutación adecuados como los routers, los cuales encaminan la información desde una línea de entrada a una línea de salida determinada.

- Contar con interfaces adecuadas para el enlace. Estas interfaces están encargadas de permitir el acople entre estaciones finales y líneas de transmisión. Las mas utilizadas son RS-232 (de 25 Y 9 Pines) y V.35.

- Cumplir con los estándares que son definidos y manejados por autoridades mundialmente reconocidas como son: ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) [11], ISO (Intemational Organization for Standardization) [12], IETF

(Intemet Engineering Task Force) [13], EIA (Electronic Industries Association) [14].

- Necesita contar con protocolos que manejen un lenguaje estándar para lograr la comunicación entre distintos dispositivos de tal forma que entre ellos puedan enviar y recibir datos coherentemente.

Dentro de los protocolos más utilizados por las redes WAN se encuentra el HDLC[2] y el PPP[3]. Estos protocolos tienen como funciones principales: Lograr la atención de las otras partes en la comunicación, proporcionar un indicador constante de que los datos están siendo recibidos y comprendidos, o bien sea todo lo contrario, solicitar la retransmisión de los datos erróneos, iniciar el procedimiento de recuperación si aparecen datos y proporcionar una forma aceptable de concluir una transmisión para garantizar que todas las partes han terminado.

1.2. CONCEPTOS BÁSICOS SOBRE FILTRADO DE PAQUETES

El filtrado de paquetes se utiliza como un medio para reducir la carga de la red y se puede ejecutar teniendo en cuenta las reglas de enrutamiento. Éstas pueden determinar diversas condiciones por las cuales un paquete puede o no pasar a través del filtro determinado. Un ejemplo de estas reglas puede ser descartar paquetes cuyo TTL ha llegado a cero, o paquetes con un control de errores erróneos, o simplemente tramas de broadcast. También se puede presentar el caso que existan reglas para el análisis del protocolo utilizado (TCP, UDP, ICMP...), las direcciones fuente y destino, o el puerto fuente y destino. Además de estas aplicaciones, el filtrado de paquetes se puede utilizar para implementar diferentes políticas de seguridad en una red; para así

[2] High-Level Data Link Control. Ver referencia Bibliográfica [2]
[3] Point to Point Protocol. Ver RFC 1661

lograr evitar el acceso no autorizado a ella, pero manteniendo intactos los accesos autorizados.

El funcionamiento de este filtro se basa en el análisis de la cabecera de cada paquete, y en función de las reglas establecidas de antemano, la trama es bloqueada o se le permite seguir su camino. Además de la información de cabecera de las tramas, algunas implementaciones de filtrado permiten especificar reglas basadas en la interfaz del router por donde se ha de reenviar el paquete, y también en la interfaz por donde ha llegado hasta nosotros.

2. CREACIÓN DE UN ENLACE WAN PUNTO A PUNTO

El primer paso de éste trabajo requiere implementar un enlace WAN punto a punto. Para ello se utilizaron tanto elementos de hardware como de software. Los elementos principales de Hardware que se requieren para el enlace son un par de adaptadores WAN, en este caso de marca SyncLink[4] que se utilizaron en computadores con sistema operativo Linux (Para éste proyecto se utilizó Red Hat Linux).

El primer paso a seguir fue la instalación física de los adaptadores WAN en los computadores, para lo cual cada tarjeta fue insertada en una de las ranuras de expansión PCI del computador. En este punto se utilizó el estándar RS-232 para la transmisión, por lo que se deben mantener los jumpers de la tarjeta en el lugar que corresponde al estándar mencionado. En el momento en el que se arranca Linux, se despliega un pantallazo (generalmente de color rojo) indicando que la herramienta de detección de hardware de Linux (Kudzu) ha encontrado un nuevo dispositivo que debe ser adicionado al sistema.

Seguidamente se procedió con la instalación del software que requiere la tarjeta para su funcionamiento. Existen dos formas de instalar los drivers para las tarjetas, pero una de ellas no provee soporte para realizar la configuración IP del enlace; Se escogió la forma más completa y a la vez más compleja (Debido a que es necesario compilar el kernel). Después de seguir el procedimiento para la instalación de los drivers con soporte para configuración IP, se procede con la instalación de las utilidades (mgslutil y mgsltest)[5].

Teniendo las utilidades y los drivers instalados, se procede a realizar una

[4] Para mayor información de hardware sobre los adaptadores referirse al anexo C
[5] Para una descripción detallada del proceso de instalación de los drivers y las utilidades referirse al anexo D

verificación de conectividad básica que consiste en efectuar un par de pruebas de loopback. El primer paso para realizar estas pruebas es cargar los drivers, para ello se utilizará la manera más sencilla que es mediante la ejecución del script que provee el fabricante llamado "load-drivers.sh". Desde un terminal (ubicados en la carpeta de distribución de Synclink) fue utilizado de la siguiente manera[6]:

[Synclink]# **./ load-drivers.sh /dev/ttySL0**

Después de activar los drivers y antes de realizar las pruebas es necesario configurar los parámetros internos de las tarjetas que serán utilizadas, para lo cuál se utiliza la herramienta "mgslutil". La primera prueba es denominada de loopback interno, la cual nos permite comprobar que las tarjetas se encuentran en funcionamiento. La configuración que se utilizó fue la siguiente: Encapsulamiento Cisco HDLC, codificación nrz, interfaz RS-232, loopback interno, y data rate 1,5 Mbps. Desde un terminal se utilizó mgslutil de la siguiente manera:

[Synclink]# **./mgslutil /dev/ttySL0 cisco rs232 hdlc nrz +loopback clock 1500000**

Después se utilizó "mgsltest" para realizar la transmisión, y se obtuvo un 100% de eficiencia de transmisión (ningún paquete perdido). Desde un terminal se ejecutó de la siguiente manera:

[Synclink]# **./Mgsltest /dev/ttySL0 size 1024 count 100**

La segunda prueba es denominada de loopback externo, y consta de dos

[6] Para mayor información sobre el manejo de las utilidades de manejo de drivers y de conectividad referirse al anexo F

partes. La primera parte utiliza una sola tarjeta y un conector de loopback externo (lo provee el fabricante) acoplado a su interfaz de salida (DB 25), la cual permite determinar que la tarjeta envía y recibe paquetes correctamente. La configuración que se utilizó fue la siguiente: Encapsulamiento Cisco HDLC, codificación nrz, interfaz RS-232, loopback externo, y data rate 115200 bps. Desde un terminal se utilizó mgslutil de la siguiente manera:

[Synclink]# **./mgslutil /dev/ttySL0 cisco rs232 hdlc nrz -loopback clock 115200**

Después se utilizó "mgsltest" para realizar la prueba, encontrándose un 100% de eficiencia de transmisión (ningún paquete perdido). Desde un terminal se ejecutó de la siguiente manera:

[Synclink]# **./Mgsltest /dev/ttySL0 size 1024 count 100**

La segunda parte de las pruebas de loopback externo utiliza las dos tarjetas, una de ellas se configura en el modo normal de transmisión (master) y la segunda tarjeta en el modo de escucha y reenvía (slave). Las dos tarjetas deben conectarse mediante un cable de Null-Modem[7] (DB25 hembra - DB25 hembra), teniendo en cuenta que en el extremo del cable marcado como "clock" se debe configurar el dispositivo maestro (master), y en el otro extremo el dispositivo esclavo (slave).

Esta prueba nos permite además de verificar el correcto funcionamiento de las tarjetas, el funcionamiento del cable de transmisión. La configuración que se utilizó fue la siguiente: Encapsulamiento Cisco HDLC, codificación nrz, interfaz RS-232, loopback externo, y data rate 115200 bps. Desde un terminal se utilizó mgslutil de la siguiente manera:

[7] En el anexo C se muestra el pinout para este cable

[Synclink]# **./mgslutil /dev/ttySL0 cisco rs232 hdlc nrz -loopback clock 115200**

Después se utilizó "mgsltest" para realizar la prueba, obteniendo un 100% de eficiencia de transmisión (ningún paquete perdido). Desde un terminal se utilizó de la siguiente manera:

[Synclink]# **./Mgsltest /dev/ttySL0 size 1024 count 100**

El siguiente paso a seguir consiste en la activación de la interfaz de red de la tarjeta WAN[8]. Para ello se utilizó la utilidad "start-cisco.sh" que provee el fabricante, de la siguiente manera:

[Synclink]# **./start-cisco.sh /dev/ttySL0**

Los parámetros de red se configuran directamente editando este archivo (start-cisco.sh). Se configuró un enlace punto a punto entre los dos computadores que tienen las tarjetas WAN como se muestra en la figura 2. En este escenario, los computadores se están comunicando por medio de un cable de Null – Modem, teniendo en cuenta que en el extremo del cable marcado como "clock" se debe configurar el DCE, y en el otro extremo el DTE.

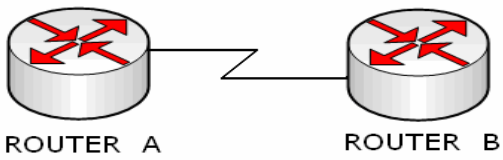

Figura 2. Conexión Punto a Punto usando los computadores que tienen las tarjetas WAN como routers

[8] Para una descripción detallada del proceso de activación de la interfaz de red, referirse al anexo F

Entre los enrutadores se estableció una red con dirección IP 200.0.0.0 y máscara de red 255.255.255.0, configurando el Router A (PC) como DCE de la siguiente manera:

ROUTER A (DCE)	*ROUTER B (DTE)*
IPADDR=200.0.0.1	IPADDR=200.0.0.2
NETMASK=255.255.255.0	NETMASK=255.255.255.0
POINTOPOINT=200.0.0.2	POINTOPOINT=200.0.0.1
GENCLOCK=112500	GENCLOCK=0

Se puede verificar fácilmente que el enlace está activado realizando un ping entre los dos equipos. Por ejemplo, se realizó un ping desde el Router A (PC) hacia la dirección IP 200.0.0.2, obteniendo 100% de paquetes transmitidos.

El siguiente paso es configurar un escenario de red como el que se muestra en la figura 3; En la cual se tienen dos nubes que simbolizan redes LAN que deben poder comunicarse entre ellas a través del enlace WAN.

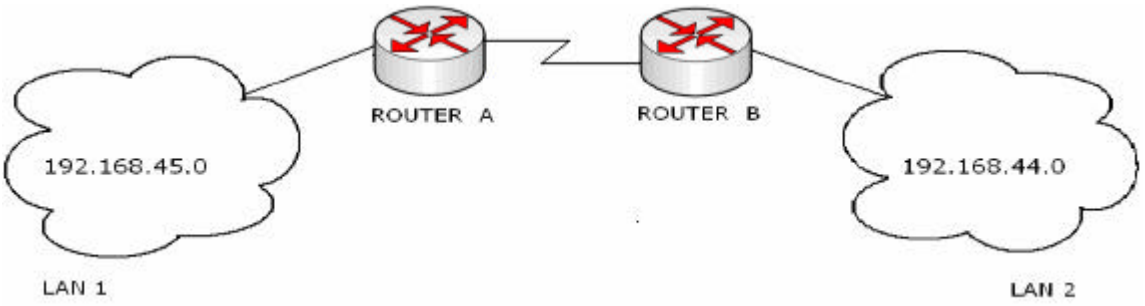

Figura 3. Escenario de conectividad entre dos redes LAN conectándose a través de una red WAN

Se implementaron las nubes LAN de la forma más sencilla, la cual es con un computador (utilizando su interfaz de red Ethernet). Teniendo ya configurado y activado el enlace punto a punto, hace falta agregar las rutas que los Routers deben seguir para alcanzar las LAN, para ello se agregarán las rutas necesarias

para cada enrutador de tal forma que cada uno de ellos conozca una forma de acceder a las redes LAN. Debido a que los routers (PC) solamente presentan dos interfaces de red, no es necesario definir un protocolo de enrutamiento como OSPF o RIP; Por el contrario, esto solo traería más tráfico y más carga para los enrutadores, más desgaste de sus recursos y desmejoraría el rendimiento final. Este tipo de red recibe el nombre de red de conexión única.

Se definió que la dirección de red correspondiente a LAN 1 es 192.168.45.0 con máscara 255.255.255.0 y para la LAN 2 es 192.168.44.0 con la misma máscara de red. Entonces en el Router A se debe eliminar el gateway por defecto que tenga para posteriormente agregar el nuevo gateway[9], y seguidamente es necesario activar el reenvío de paquetes (ip_forward) para que el computador pueda funcionar como router (enviar paquetes de una interfaz a otra). Desde un terminal (cualquier ubicación) se realizó lo siguiente:

```
[root]# route delete default
[root]# route add default gw 200.0.0.2
[root]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Para el Router B no se puede definir como gateway por defecto la dirección del enrutador A, pues se presenta un riesgo muy elevado de que se generen loops de enrutamiento, lo cuál sobrecargaría el enlace. Entonces para el enrutador B se definirá la ruta estática para encontrar la red que no conoce (LAN 1), de la siguiente manera:

```
[root]# route delete default
[root]# route add -net 192.168.45.0 netmask
        255.255.255.0 gw 200.0.0.1
[root]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

[9] Para una descripción detallada de cómo agregar rutas estáticas en linux, referirse al anexo G

Por último se deben modificar los parámetros de red de los computadores pertenecientes a las LAN y deben configurarse las interfaces ethernet de los Routers (PC). Desde el router A se configurará la dirección IP 192.168.45.1 con máscara 255.255.255.0 para su interfaz ethernet, y desde el router B 192.168.44.1 con la misma máscara[10]. La dirección IP del computador perteneciente a la red LAN 1 es 192.168.45.2 con máscara 255.255.255.0 y con gateway 192.168.45.1; y para el computador ubicado en la LAN 2 la dirección IP es 192.168.44.2 con máscara 255.255.255.0 y gateway 192.168.44.1.

Para realizar la respectiva verificación del funcionamiento del montaje se realizaron peticiones de ping desde el computador de la LAN 1 hacia cada interfaz intermedia (Routers) y hacia el computador en la LAN 2 y viceversa. El resultado siempre fue 100% de paquetes entregados.

[10] Consultar página man del comando ifconfig o manuales relacionados con redes en linux

3. DESARROLLO DE UN FIREWALL EN UN SISTEMA OPERATIVO LINUX

Un firewall es un mecanismo implementado mediante hardware y/o software que permite controlar el tráfico de datos que circula entre dos redes, imponiendo entre ellas una política de seguridad. Para que un firewall sea efectivo, todo tráfico de información que ingrese o salga de la red deberá pasar a través del mismo donde la información podrá ser examinada y posteriormente filtrada.

El sistema operativo Linux provee una serie de herramientas que permiten la implementación de un firewall en éste; ya sea a nivel de aplicaciones de usuario o a nivel del Kernel. Entre éstas últimas se presenta una alternativa de implementación mediante programación de módulos[11] denominada Netfilter Hooks[12].

Esta herramienta está basada en las librerías "Netfilter.h" y "Netfilter_ipv4.h", que en Red Hat Linux se encuentran en la ubicación completa "/usr/include/linux/". A continuación se describe la estructura, implementación y el funcionamiento de un firewall utilizando Netfilter Hooks. Realizado sobre Red Hat Linux 9.0 usando una versión de Kernel 2.4.20-8.

Existen en total cinco instantes diferentes en los que pueden ser capturados los paquetes para su evaluación y su respectivo filtrado. En esta implementación se escogieron solo dos; el primer momento para capturar es justo después de que el paquete ingresa por una interfaz dada (antes de las decisiones de enrutamiento) y el segundo es antes de entregar los paquetes por la interfaz de salida (después de las decisiones de enrutamiento). Una buena utilización

[11] Consulte referencia bibliográfica [8]
[12] Consulte referencia bibliográfica [10]

de estos instantes permite un mejor rendimiento del sistema[13].

Para poder filtrar el tráfico de paquetes, el firewall se basa en una serie de criterios denominados comúnmente como reglas de filtrado. Dichas reglas pueden utilizarse individualmente o realizando combinaciones de éstas para mayor complejidad de filtrado, lo que indica que se puede crear un filtrado muy selectivo. Las reglas que se implementaron fueron las siguientes:

1) **Filtrado por interfaz:** Tiene en cuenta por que interfaz llega o sale el paquete. Se realizó para las tres interfaces básicas del computador que funciona como firewall (Wan, Lan, y Loopback[14]). Veamos a continuación como se implementa la regla.

```
 if (strcmp(in->name, interface_WAN) == 0)
{
    return NF_ACCEPT;
}
```

Simplemente se realiza una comparación entre la interfaz por la cual entra el paquete al equipo y el nombre de la interfaz que implementará la regla (en este caso interface_WAN). Si en la comparación tiene éxito y la interfaz de entrada es la misma que la que se busca filtrar (resultado de la función strcmp = 0), entonces se realiza la acción de filtrado que se defina dentro de la función “if”; Se puede observar que todos los paquetes son aceptados (NF_ACCEPT). En éste firewall, se definieron dos nombres de interfaces, que son, “interface_WAN = mgsl0” y “interface_LAN = eth0”.

NOTA: En el firewall realizado se tiene esta regla implementada como parte de la estructura básica, permitiendo escoger la interfaz en la cual será aplicada la regla.

[13] Por sistema se refiere a la combinación de PC (Hardware) y Firewall (Software)
[14] La interfaz de loopback se implementó para verificar el funcionamiento de las reglas de filtrado sin utilizar otros equipos.

2) **Filtrado por dirección IP:** Tiene en cuenta la dirección IP (puede ser tanto IP fuente como IP destino) del paquete. Veamos a continuación como se implementa la regla.

```
struct sk_buff *sb = *skb
if (!sb ) return NF_ACCEPT;
if (!(sb->nh.iph)) return NF_ACCEPT;
if (sb->nh.iph->daddr == *(unsigned int *)direccionIP_1)
{
      return NF_DROP;
}
else
{
      return NF_ACCEPT;
}
```

El campo “daddr” que indica dirección IP destino, puede ser sustituido por “saddr” para realizar el filtrado teniendo en cuenta la dirección IP fuente. Simplemente se realiza una comparación entre la dirección IP fuente o destino del paquete entrante y la dirección IP con la que se desea realizar el filtrado. Si en la comparación tiene éxito y la dirección IP del paquete es la misma que se desea filtrar (resultado de la función strcmp = 0), entonces se realiza la acción de filtrado que se defina dentro de la función “if”; Se puede observar que todos los paquetes son descartados (NF_DROP).

3) **Filtrado por protocolo:** Es posible filtrar teniendo en cuenta los protocolos de capa de transporte UDP ó TCP y teniendo en cuenta si es un paquete ICMP. Veamos a continuación como se implementa la regla.

```
struct sk_buff *sb = *skb;
if (!sb ) return NF_ACCEPT;
if (!(sb->nh.iph)) return NF_ACCEPT;
if (sb->nh.iph->protocol == 17)  //Cambiar por: 6, 17, 1
{
      return NF_DROP;   //Elimina paquetes
}
else
{
      return NF_ACCEPT;
}
```

En esta regla de filtrado, se realiza una sencilla comparación entre el campo de protocolo de la cabecera del paquete y el respectivo protocolo que se desea filtrar, que está dado por el número 17 para el protocolo UDP, 6 para el protocolo TCP, y 1 para el protocolo ICMP. Como se ve, la regla está predefinida para eliminar los paquetes que cumplan con el protocolo seleccionado y para aceptar los demás.

4) **Filtrado por puerto:** Tiene en cuenta el puerto de capa de transporte (puede ser puerto fuente o puerto destino) que utilice el paquete para su transmisión. Veamos a continuación como se implementa la regla.

```
struct sk_buff *sb = *skb;
struct udphdr *uhead;
if (!sb ) return NF_ACCEPT;
if (!(sb->nh.iph)) return NF_ACCEPT;
if (sb->nh.iph->protocol == 17) //Escoger protocolo
{
    uhead = (struct udphdr *)(sb->data + (sb->nh.iph->ihl * 4));
    //Filtrado por Puerto, escoja “dest” o “source”
    if ((uhead->dest) == *(unsigned short *)port_1)
    {
        return NF_DROP; //Elimina el paquete
    }
    else
    {
        return NF_ACCEPT;
    }
}
```

Esta regla de filtrado, debe aplicarse como complemento de la regla de filtrado por protocolo, pero solo para TCP y UDP, con el fin de poder definir la cabecera de la capa de transporte para posteriormente obtener de ésta el valor del puerto ya sea fuente o destino. La regla trabaja realizando una comparación entre el puerto de la cabecera del segmento (capa de transporte) y el respectivo puerto que se desea filtrar, que está dado por la palabra “dest” para puerto destino y “source” para puerto fuente. Como se ve, la regla está predefinida para eliminar los paquetes que cumplan con el puerto seleccionado y para aceptar los demás.

5) **Filtrado de peticiones de Ping Indeseables:** Se puede activar para descartar cualquier petición de ping que se haga hacia dentro o fuera de la red o hacia las interfaces del firewall. Tiene la posibilidad de aceptar las peticiones de un host administrador, el cual se vea en la necesidad de verificar el estado de las interfaces del firewall. Veamos a continuación como se implementa la regla.

```
struct icmphdr *icmp;
struct sk_buff *sb = *skb;
if (!sb ) return NF_ACCEPT;
if (!(sb->nh.iph)) return NF_ACCEPT;
icmp = (struct icmphdr*)(sb->data + (sb->nh.iph->ihl * 4));
if (sb->nh.iph->protocol == 1)  //paquetes ICMP
{
      if (icmp->type ==8) //Escoge peticiones ping
      {
      //Acepta las peticiones ping del administrador
            if (sb->nh.iph->saddr == *(unsigned int *)direccionIP_A)
            {
                  return NF_ACCEPT;
            }
            //Elimina todas las peticiones ping indeseables
            else
            {
                  return NF_DROP;
            }
      }
}
```

En esta regla, es necesario definir primero la cabecera de ICMP, la cuál será examinada en busca de paquetes de peticiones de ping. Lo primero que se puede observar es que se está utilizando parte de la regla de filtrado por protocolo (ICMP =1), luego se realiza una comparación entre el tipo de paquete ICMP que llegó al firewall y el que se desea filtrar, que es el número 8 que corresponde a peticiones ping. En este punto se da la posibilidad de aceptar las peticiones ping del administrador, comparando una dirección autorizada con la dirección IP fuente del paquete entrante. Como se puede ver, la regla está predefinida para aceptar los paquetes cuya dirección IP fuente sea la del administrador y para descartar los demás.

6) **Limitador de paquetes por segundo:** Limita a los hosts para que transmitan por encima de un ancho de banda dado, el cual está determinado por el número máximo de paquetes/s que pueden ser transmitidos hacia dentro o fuera de la red. Veamos a continuación como se implementa la regla.

```
if_counter++;     //incrementa el contador
struct timeval ahora;
do_gettimeofday(&ahora);  //Obtiene la Hora exacta

if (!tiemporal) //Se usa al cargar la 1 vez el modulo)
{
      tiemporal=ahora.tv_sec;//cada 1 segundos
      temp_usec=ahora.tv_usec;
      return NF_ACCEPT;
}
else  //Para los demás paquetes
{
      if (ahora.tv_sec == tiemporal)
      {
            if ( if_counter <= 500 )      //Número de paquetes/s
            {
                  return NF_ACCEPT;
            }
            else
            {
                  return NF_DROP;
            }
      }
      if (ahora.tv_sec == (tiemporal +1))
      {
            dif_usec = (1000000 - temp_usec) + ahora.tv_usec;
            if (dif_usec <= 999999)
            {
                  if ( if_counter <= 500 )    //Numero de pkt/s
                  {
                        return NF_ACCEPT;
                  }
                  else
                  {
                        return NF_DROP;
                  }
            }
      }
      else
      {     tiemporal=ahora.tv_sec;//cada 1 segundos
            temp_usec=ahora.tv_usec;
            if_counter = 1;   //Reset al contador
            return NF_ACCEPT;
      }
}
```

Se utiliza una variable llamada “if_counter” para realizar un conteo del número de paquetes que son examinados. Además se utiliza la función “do_gettimeofday” definida en la librería “linux/time.h” para poder hacer una verificación precisa del instante en el cuál llega un paquete. Para modificar la cantidad de paquetes/s que puede admitir es necesario modificar el número que aparece en la línea que dice “if_counter <= 500 ”, debe observarse que esta línea aparece dos veces en el programa, entonces es preciso modificar el número dos veces para que la regla funcione correctamente.

En el anexo A se muestra el programa base para el filtrado de paquetes, sin aplicar ninguna regla específica. Para aplicar algún tipo de filtrado, se deben agregar las líneas de código al programa teniendo en cuenta la estructura de las reglas que se presenta en este capítulo. La mayoría de las reglas pueden ser fácilmente modificadas (cambiar protocolo, elegir si se aceptan o se descartan los paquetes, etc.) para suplir las necesidades de filtrado de la red. Para utilizar el firewall, debe compilarse el programa y seguidamente debe instalarse el módulo resultante[15]. Para facilitar éste proceso se realizó un programa en “bash”, el cuál se presenta con detalle en el anexo A.

Es recomendable que las reglas de filtrado sean aplicadas sobre la parte del programa en la que evalúa los paquetes entrantes; debido a que los paquetes salientes ya han pasado por el proceso de enrutamiento del equipo para seleccionar la interfaz de salida apropiada, causando una mayor demora en el procesamiento de paquetes dentro del Firewall.

[15] Para obtener información sobre el manejo de los módulos del kernel, ver referencia bibliográfica [8]

4. PRUEBAS REALIZADAS Y RESULTADOS OBTENIDOS.

Para esta etapa se plantearon diversas pruebas que tienen como fin realizar el análisis de desempeño del enlace WAN. Para este propósito se plantean 2 clases de pruebas: Pruebas del enlace entre las estaciones sin firewall y Pruebas del enlace entre las estaciones con firewall.

Para ambas pruebas se requiere de un montaje común el cual se muestra en la figura 4.

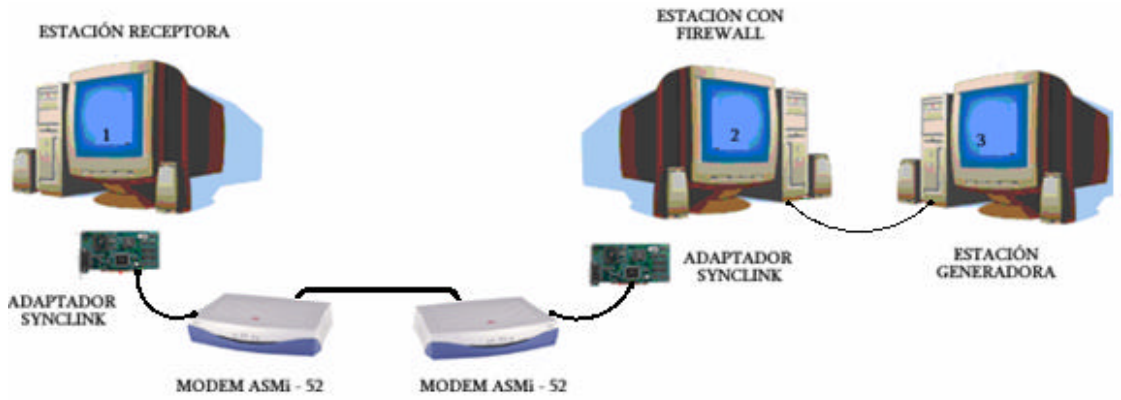

Figura 4. Esquema de red para probar el funcionamiento del firewall

En este esquema se puede observar lo siguiente:

- En la estación receptora y en la estación con firewall se encuentran instalados los adaptadores de Synclink.
- La estación generadora contiene la herramienta software MGEN la cual está encargada de la generación de tráfico en la red.
- La estación receptora contiene la herramienta software DREC la cual está encargada de la detección de tráfico en la red.

- La estación 2 contiene el firewall el cual es ejecutado en la segunda parte de las pruebas.
- Los modems ASMi-52[16] fueron configurados para transmitir a una tasa de 2,048 Mbps.

MGEN[17] es una herramienta software de libre distribución creada por la NRL[18] con el fin de realizar pruebas de transmisión para caracterizar redes. Estos tests son realizados gracias a la generación de tráfico sobre las redes que se están estudiando y así poder analizar el comportamiento y la respuesta de la red frente a estas situaciones. Las pruebas se generan usando trafico UDP. El set de herramientas genera patrones de tráfico en tiempo real de tal manera que la red pueda ser cargada de diversas formas. El tráfico generado puede ser posteriormente recibido y utilizado para realizar análisis. La herramienta encargada de la recepción y presentación del tráfico en la estación receptora es DREC [9].

A continuación se presentan las pruebas y sus respectivos resultados. Para las pruebas se presenta un formato que consta del nombre general de la prueba a la cual pertenece (con o sin tráfico), el número de la prueba correspondiente, una breve descripción, los objetivos y finalmente los resultados obtenidos.

4.1 PRUEBAS SIN FIREWALL

El objetivo general de estas pruebas es verificar el buen funcionamiento del enlace WAN y poner a prueba su capacidad.

---

[16] Para obtener información sobre su configuración consulte el manual de instalación y operación. Ver referencia bibliográfica [15]

[17] Por sus siglas en inglés *Multi - Generator*

[18] Por sus siglas en inglés *Naval Research Laboratory*

Prueba No. 1

El objetivo de esta prueba es determinar el número de paquetes máximo que puede ser transmitido sin que se presenten pérdidas en el receptor (throughput), utilizando el tamaño de paquete mínimo que puede generar la herramienta Mgen. La tabla 1 muestra los resultados obtenidos al realizar el envío de paquetes desde la estación generadora hacia la estación receptora, utilizando un entorno uniforme, con un flujo constante de paquetes/s para cada muestra y con un tamaño fijo de 37 bytes/paquete + 32 bytes de cabecera. Los valores de los resultados se encuentran en paquetes, tanto los enviados y los recibidos.

TAMAÑO (Bytes/paquete)	**RATA** (pps[19])	**DURACIÓN** (Segundos)	**RESULTADO** Enviado –Recibido
69	1	1	1-1
69	100	1	100-100
69	500	1	500-500
69	800	1	800-800
69	900	1	900-900
69	1000	1	1000-999
69	2000	1	2000-1209
Número Máximo de Paquetes/s transmitidos sin errores: **986**			

Tabla 1. Valor de Throughput para tamaño de paquete mínimo en prueba de transmisión sin Firewall

Para el tamaño de paquete mínimo (69 bytes) se presentan pérdidas a partir de una tasa de 986 pps. De acuerdo con este valor, se puede extraer el ancho de banda real que está utilizando para la transmisión:

$$(986 \text{ paquete/s}) \times (69 \text{ bytes/paquete}) \times (8 \text{ bits/byte}) = 544{,}272 \text{ Kbps}$$

El valor máximo que alcanza es de 667,368 Kbps, utilizando 1209 pps que es la máxima tasa de transmisión que se obtuvo.

[19] Paquetes por segundo

Prueba No. 2

El objetivo de esta prueba es determinar el número de paquetes máximo que puede ser transmitido sin que se presenten pérdidas en el receptor. La tabla 2 muestra los resultados obtenidos al transmitir paquetes desde la estación generadora hacia la estación receptora, utilizando un entorno uniforme, con un flujo constante de paquetes/s para cada muestra y con un tamaño fijo de 256 bytes/paquete + 32 bytes de cabeceras. Los valores de los resultados se encuentran en paquetes, tanto los enviados y los recibidos.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**RESULTADO** Enviado –Recibido
288	500	1	500-500
288	800	1	800-800
288	820	1	820-820
288	830	1	830-822
288	850	1	850-822
288	986	1	986-825
Número Máximo de Paquetes/s transmitidos sin errores: **822**			

Tabla 2. Valor de Throughput al aumentar el tamaño de paquete en prueba de transmisión sin Firewall

Para el tamaño de paquete utilizado (288 bytes) se presentan pérdidas a partir de una tasa de 822 pps. De acuerdo con este valor, se puede extraer el ancho de banda real que está utilizando para la transmisión:

$$(822 \text{ paquete/s}) \times (288 \text{ bytes/paquete}) \times (8 \text{ bits/byte}) = 1{,}893 \text{ Mbps}$$

El valor máximo que alcanza es de 1,9 Mbps, utilizando 825 pps que es la máxima tasa de transmisión que se obtuvo.

Se realizaron más pruebas aumentando el tamaño del paquete, transmitiendo durante 1 segundo, con el fin de observar la respuesta del enlace. El resultado fue que se logró una mejor utilización del ancho de banda ($\approx$2 Mbps); Además se observó que al aumentar el tamaño del paquete, la estación generadora se

veía en la obligación de fragmentar los paquetes para su transmisión (El tamaño máximo del paquete sin fragmentar es 1514 Bytes con cabecera incluida).

Prueba No. 3

El objetivo de esta prueba es determinar el número de paquetes máximo que puede ser transmitido sin que se presenten pérdidas en el receptor, además de observar el comportamiento del enlace cuando la generación de paquetes se mantiene por un tiempo mayor. La tabla 3 muestra los resultados obtenidos al transmitir paquetes desde la estación generadora hacia la estación receptora, utilizando un entorno uniforme, con un flujo constante de paquetes/s para cada muestra y con un tamaño fijo de 37 bytes/paquete + 32 bytes de cabecera.

Se realiza la prueba con un tamaño de paquete constante y un mismo periodo de duración (60 segundos), y se varía el número de paquetes por segundo. Los valores de los resultados se encuentran en paquetes, tanto los enviados y los recibidos. Se puede observar una ligera pérdida de paquetes en comparación con la transmisión durante 1segundo.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**RESULTADO** Enviado –Recibido
69	986	60	59161 -59143
69	950	60	57000 -56980
69	900	60	54000 -53983
69	800	60	48000 -47988
69	700	60	42000 -41980
69	650	60	39000 -38995
69	600	60	36001 -35997
69	550	60	35000 -32998
69	520	60	31201 - 31201
69	500	60	30000 -30000
69	500	300	150000 – 149996
Número Máximo de Paquetes/s transmitidos sin errores: **520**			

Tabla 3. Valor de Throughput para tamaño de paquete mínimo aumentando la duración en prueba de transmisión sin Firewall

Prueba No. 4

Con flujo de paquetes no uniforme (simulación de tráfico real). El objetivo de esta prueba es ver la posibilidad de encontrar el número de paquetes máximo que puede ser transmitido sin presentar pérdidas en el receptor. La tabla 4 muestra los resultados obtenidos al realizar el envío de paquetes desde la estación generadora hacia la estación receptora. Se realiza la prueba con un tamaño de paquete constante y un mismo periodo de duración (1 segundo), y se varía el número de paquetes por segundo.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**RESULTADO** Enviado –Recibido
69	1	1	4 - 4
69	100	1	100-100
69	200	1	204-204
69	300	1	287-287
69	400	1	418-418
69	450	1	422-422
69	500	1	531-531
69	800	1	832-815
69	900	1	891 -865
Número Máximo de Paquetes/s transmitidos sin errores: **450**			

Tabla 4. Valor de Throughput para tamaño de paquete mínimo en prueba con simulación de tráfico real sin Firewall

De acuerdo con los resultados obtenidos se nota que al introducir tráfico real (flujo y tamaño del paquete variables a través del tiempo) en la generación de paquetes no se mantienen los patrones que en los casos de generación uniforme se habían observado. Además se nota que el valor límite en el que no se produce pérdida de paquetes es de 450 pps. Debido a que el nivel de tráfico es variable, es un tanto difícil determinar a que se debe el error en la transmisión de los paquetes; Se puede observar una ligera disminución de la eficiencia de transmisión.

Prueba No. 5

El objetivo de esta prueba es determinar el número de paquetes máximo que puede ser transmitido sin que se presenten pérdidas en el receptor cuando en el generador se incluye tráfico real en la emisión, además de observar el comportamiento del enlace cuando la generación de paquetes se mantiene por un tiempo mayor. La tabla 5 muestra los resultados obtenidos al transmitir paquetes desde la estación generadora hacia la estación receptora, utilizando paquetes con tamaño fijo de 256 bytes/paquete + 32 bytes de cabecera.

De acuerdo con los resultados obtenidos se concluye que al introducir tráfico se disminuye le valor máximo de pps que pueden ser transmitidos sin error. En este caso el valor disminuyó de 822 a 390 pps. Se puede observar una ligera disminución de la eficiencia de transmisión, y debido a que el nivel de tráfico es variable, es un tanto difícil determinar las posibles fuentes de error.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**RESULTADO** Enviado –Recibido
288	350	1	357-357
288	380	1	387-387
288	390	1	393-392
288	395	1	359-359
288	400	1	404-403
288	500	1	501-498
288	820	1	819-775
288	830	1	873-792
Número Máximo de Paquetes/s transmitidos sin errores: **390**			

Tabla 5. Valor de Throughput al aumentar el tamaño de paquete en pruebas con simulación de tráfico real sin Firewall

Se realizaron una serie de pruebas extra transmitiendo con tráfico real y con un tamaño de paquete mayor (cercano a 1Kbyte), pero no se encontró información concluyente excepto que el rendimiento y la capacidad del enlace disminuyen debido a las condiciones adversas de la transmisión.

Prueba No. 6

Encontrar el número de paquetes máximo que puede ser transmitido sin presentar pérdida de paquetes en el receptor cuando la estación generadora emite paquetes con simulación de tráfico real en la emisión. La tabla 6 muestra los resultados obtenidos al transmitir paquetes desde la estación generadora hacia la estación receptora, utilizando un tamaño fijo de 37 bytes/paquete + 32 bytes de cabeceras. Se realiza la prueba con un tamaño de paquete y duración constantes, y se varía el número de paquetes por segundo.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**RESULTADO** Enviado –Recibido
69	986	60	59340 -56335
69	450	60	27002 -26996
69	400	60	23784 -23782
69	350	60	21313 -21312
69	340	60	20358 -20358
69	300	60	18112 -18112
Número Máximo de Paquetes/s transmitidos sin errores: **345**			

Tabla 6. Valor de Throughput para tamaño de paquete mínimo aumentando la duración en prueba de transmisión con simulación de tráfico real sin Firewall

Se puede observar que el valor límite en el que no se producen pérdidas es de 345 pps. Se puede observar una ligera disminución de la eficiencia de transmisión, y debido a que el nivel de tráfico es variable, es un tanto difícil determinar las posibles fuentes de error.

4.2 PRUEBAS CON FIREWALL

El objetivo general de estas pruebas es observar el nivel de pérdidas en el enlace WAN dadas por la activación de un Firewall en uno de los routers (PC).

Prueba No. 7: Bloqueo del Puerto destino.

El objetivo de esta prueba es verificar la respuesta del enlace cuando se utiliza

el Firewall activando solamente la regla que bloquea o permite el paso de paquetes teniendo en cuenta el puerto destino de capa de transporte.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**Puerto utilizado**	**RESULTADO** Enviado –Recibido
69	1	1	5000	1-1
69	100	1	5000	100-100
69	500	1	5000	500-500
69	800	1	5000	800-800
69	900	1	5000	900-900
69	986	1	5000	986-985
69	987	1	5000	986-985
69	1000	1	5000	1000-1001
69	1200	1	5000	1200-1175
69	1	1	5001	1-0
69	100	1	5001	100-0
69	500	1	5001	500-0
69	800	1	5001	800-0
69	900	1	5001	900-0
69	986	1	5001	986-0
69	987	1	5001	987-0
69	1000	1	5001	1000-0
69	1200	1	5001	1200-0
69	986	60	5001	59101-0
69	986	300	5001	295501-0

Tabla 7. Resultados obtenidos en la estación receptora para pruebas de filtrado con la opción bloqueo del Puerto destino activada

La tabla 7 muestra los resultados obtenidos al realizar el envío de paquetes desde la estación generadora hacia la estación receptora pasando por la estación Firewall. Se realiza la prueba con un tamaño de paquete y duración constantes (Exceptuando las 2 últimas pruebas), se varía el número de paquetes por segundo y se cambia en el filtro el valor del puerto destino para verificar el funcionamiento de la regla.

Se obtienen resultados satisfactorios en cuanto a rendimiento del enlace cuando se bloquea un puerto distinto al puerto destino que tengan los paquetes (Se tiene bloqueado el puerto 5001). En el caso que se envíen paquetes por el puerto bloqueado, entonces la regla funciona correctamente y descarta todos los paquetes, es decir el número de paquetes recibido es cero.

Prueba No. 8: Bloqueo de dirección IP destino.

El objetivo de esta prueba es verificar la respuesta del enlace cuando se utiliza el Firewall activando solamente la regla que bloquea o permite el paso de paquetes teniendo en cuenta la dirección IP destino que posean. La tabla 8 muestra los resultados obtenidos al realizar el envío de paquetes desde la estación generadora hacia la estación receptora pasando por la estación que contiene activado el Firewall. Se realiza la prueba con un tamaño de paquete y duración constantes (Exceptuando las 2 últimas pruebas), se varía el número de paquetes por segundo y se cambia en el filtro el valor de la dirección IP destino.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**Dirección IP destino**	**RESULTADO** Enviado–Recibido
69	1	1	200.0.0.1	1-1
69	100	1	200.0.0.1	100-100
69	500	1	200.0.0.1	500-500
69	800	1	200.0.0.1	800-800
69	900	1	200.0.0.1	900-900
69	986	1	200.0.0.1	986- 985
69	987	1	200.0.0.1	986- 985
69	1000	1	200.0.0.1	1000-1001
69	1200	1	200.0.0.1	1200-1175
69	1	1	192.168.45.43	1-0
69	100	1	192.168.45.43	100-0
69	500	1	192.168.45.43	500-0
69	800	1	192.168.45.43	800-0
69	900	1	192.168.45.43	900-0
69	986	1	192.168.45.43	986-0
69	987	1	192.168.45.43	987-0
69	1000	1	192.168.45.43	1000-0
69	1200	1	192.168.45.43	1200-0
69	986	60	192.168.45.43	59101-0
69	986	300	192.168.45.43	295501-0

Tabla 8. Resultados obtenidos en la estación receptora para pruebas de filtrado con la opción bloqueo de dirección IP destino activada

Se obtienen resultados satisfactorios en cuanto a rendimiento del enlace cuando se bloquea una dirección IP distinta a la dirección IP destino que tengan los paquetes (Se tiene bloqueada la IP 192.168.45.43). En el caso que

se envían paquetes a la dirección bloqueada, entonces la regla funciona correctamente y descarta todos los paquetes.

Se realizaron pruebas adicionales para comprobar el funcionamiento de la regla que permite bloquear ciertas direcciones IP fuente, y los resultados obtenidos fueron los mismos que para el caso de la IP destino.

Prueba Número 9: Bloqueo de protocolo UDP de la capa de transporte.
El objetivo de esta prueba es verificar la respuesta del enlace cuando se utiliza el Firewall activando solamente la regla que bloquea o permite el paso de paquetes teniendo en cuenta el protocolo de capa de transporte que se utilice para la transmisión.

TAMAÑO (Bytes/paquete)	**RATA** (pps)	**DURACIÓN** (Segundos)	**Protocolo UDP**	**RESULTADO** Enviado–Recibido
69	1	1	NO	1,-1
69	100	1	NO	100-100
69	500	1	NO	500-500
69	800	1	NO	800-800
69	900	1	NO	900-900
69	986	1	NO	986- 985
69	987	1	NO	986- 985
69	1000	1	NO	1000-1001
69	1200	1	NO	1200-1175
69	1	1	SI	1-0
69	100	1	SI	100-0
69	500	1	SI	500-0
69	800	1	SI	800-0
69	900	1	SI	900-0
69	986	1	SI	986-0
69	987	1	SI	987-0
69	1000	1	SI	1000-0
69	1200	1	SI	1200-0
69	986	60	SI	59101-0
69	986	300	SI	295501-0

Tabla 9. Resultados obtenidos en la estación receptora para pruebas de filtrado con la opción bloqueo de protocolo UDP de la capa de transporte activada

La tabla 9 muestra los resultados obtenidos al realizar el envío de paquetes desde la estación generadora hacia la estación receptora pasando por la

estación Firewall. Se realiza la prueba con un tamaño de paquete y duración constantes (Exceptuando las 2 últimas pruebas), se varía el número de paquetes por segundo y se cambia en el filtro el valor de la dirección IP destino.

Se obtienen recepción de paquetes cuando el protocolo UDP no se encuentra bloqueado, de lo contrario no se presentan paquetes en la estación receptora.

Prueba No. 10: Bloqueo por número máximo de paquetes por segundo.
El objetivo de realizar esta prueba es verificar el correcto funcionamiento de la regla que permite limitar a un usuario a todo el enlace a funcionar con un limitador de flujo dado por la cantidad de paquetes/s. La tabla 10 muestra los resultados obtenidos al realizar el envío de paquetes desde la estación generadora hacia la estación receptora pasando por la estación Firewall. Se realiza la prueba con un tamaño de paquete y duración constantes (Exceptuando las 4 últimas pruebas), se varía el número de paquetes por segundo y se cambia en el filtro el número de paquetes por segundo máximo que se puede transmitir.

Se obtienen recepción de paquetes igual a los enviados cuando el nivel límite de paquetes por segundo máximos es mayor al número de paquetes utilizado en las pruebas. De lo contrario el número de paquetes recibido está determinado por el valor que presenta en el filtro, pero con un pequeño error debido a que el generador de paquetes en algunos casos transmite por más del tiempo estipulado, enviando paquetes que lograrán superar el firewall.

De todas las reglas de filtrado implementadas individualmente, ésta es la que presenta un mayor procesamiento en cuanto a líneas de código a ejecutar; sin embargo, el rendimiento del enlace no se vio afectado.

TAMAÑO (Bytes/paquete)	RATA (pps)	DURACIÓN (Segundos)	Límite de paquetes/s	RESULTADO Enviado–Recibido
69	100	1	200	100-100
69	100	1	50	100-50
69	500	1	600	500-500
69	500	1	300	500-303
69	800	1	900	800-800
69	800	1	500	800-503
69	1000	1	1200	1000-1000
69	1000	1	800	1000-805
69	986	60	1000	59101-59081
69	986	300	1000	295485-295377

Tabla 10. Resultados obtenidos en la estación receptora para pruebas de filtrado con la opción bloqueo por número máximo de paquetes por segundo activada

Prueba No. 11: Bloqueo de peticiones de ping.

El objetivo de esta prueba es verificar la respuesta presentada por el enlace cuando se utiliza el Firewall activando solamente la regla que bloquea o permite el paso de peticiones de ping hacia el firewall o a través de él.

DIRECCIÓN IP ADMINISTRADOR PREMITIDA: 198.168.45.44				
TAMAÑO (Bytes)	**RATA** (pps)	**DURACIÓN** (Segundos)	**DIRECCIÓN IP FUENTE**	**RESULTADO**
69	1	1	192.168.45.44	CONEXIÓN OK
69	100	1	192.168.45.44	CONEXIÓN OK
69	500	1	192.168.45.44	CONEXIÓN OK
69	800	1	192.168.45.44	CONEXIÓN OK
69	900	1	192.168.45.44	CONEXIÓN OK
69	986	1	192.168.45.44	CONEXIÓN OK
69	1000	1	192.168.45.44	CONEXIÓN OK
69	1200	1	192.168.45.44	CONEXIÓN OK
69	1	1	192.168.45.43	FALLÓ
69	100	1	192.168.45.43	FALLÓ
69	500	1	192.168.45.43	FALLÓ
69	800	1	192.168.45.43	FALLÓ
69	900	1	192.168.45.43	FALLÓ
69	1200	1	192.168.45.43	FALLÓ
69	986	60	192.168.45.43	FALLÓ
69	986	300	192.168.45.43	FALLÓ

Tabla 11. Resultados obtenidos en la estación receptora para pruebas de filtrado con la opción bloqueo de peticiones de ping activada

La tabla 11 muestra los resultados obtenidos al realizar el envío de paquetes

desde la estación generadora hacia la estación receptora pasando por la estación Firewall. Se realiza la prueba con un tamaño de paquete y duración constantes (Exceptuando las 2 últimas pruebas), se varía el número de paquetes por segundo y se cambia en el filtro el valor de la dirección IP no permitida.

Se responde la petición de ping cuando la dirección del equipo que emite la petición es la permitida (Administrador del Firewall), de lo contrario la petición de ping desde la estación generadora no autorizada dará como resultado un destino inalcanzable.

Prueba No. 12: Bloqueo con todas las opciones del filtro.
El objetivo de esta prueba es verificar la respuesta presentada por el enlace cuando se utiliza el Firewall activando el mayor número de reglas simultáneas del bloqueo en el filtro, de tal forma que queden unas dependiendo de las otras con el fin de generarle la mayor carga de procesamiento al enlace. La tabla 12 muestra los resultados obtenidos al realizar el envío de paquetes desde la estación generadora hacia la estación receptora. Se realiza la prueba con un tamaño de paquete y duración constantes, se varía el número de paquetes por segundo.

El firewall está configurado para que revise que el protocolo de capa de transporte sea UDP, si cumple el requisito revisa que el puerto destino de capa de transporte utilizado sea el 5000, si cumple este otro requisito revisa luego que la dirección IP destino sea la 200.0.0.1, si cumple ese requisito entonces limita a utilizar un ancho de banda de tan solo 500 paquetes/s; Dando como resultado una cadena de reglas dependientes entre sí, con una carga de procesamiento muy elevada.

TAMAÑO (Bytes)	RATA (pps)	DURACIÓN (Segundos)	PAQUETES ENVIADOS	PAQUETES RECIBIDOS
69	100	1	100	100
69	200	1	200	200
69	300	1	301	300
69	400	1	400	400
69	490	1	491	486
69	500	1	500	501
69	550	1	550	503
69	800	1	800	505
69	986	1	987	508
69	1000	1	1000	509
69	100	60	6000	5992
69	200	60	12000	11998
69	300	60	18001	17998
69	400	60	24000	23984
69	490	60	29401	29345
69	500	60	30000	29756
69	986	60	59101	23935
69	1000	60	60000	23152
160	100	1	100	100
160	200	1	200	200
160	300	1	301	300
160	400	1	400	400
160	490	1	491	450
160	500	1	500	449
160	550	1	550	505
160	800	1	800	319
160	986	1	987	508
160	1000	1	1000	506

Tabla 12. Resultados obtenidos en la estación receptora para pruebas con todas las opciones del filtro activada

Se observa que en general, aumenta un poco el nivel de pérdidas del enlace, y en algunas ocasiones se puede observar un nivel de pérdidas muy alto, especialmente al aumentar el tamaño del paquete.

4.3 ANÁLISIS DE RESULTADOS

De acuerdo con las pruebas realizadas se observó que el desempeño del enlace se ve afectado significativamente por la variación de diversos parámetros presentes de forma permanente u ocasional en el enlace. Las pruebas fueron

realizadas teniendo presente que cada una de ellas representara un conjunto de posibles situaciones a las cuales pudiera estar sometido el enlace en operación normal. Para esto se escogieron los parámetros más importantes del enlace, como lo son el tamaño de paquete a transmitir, la rata, y el tiempo de emisión de paquetes, con el fin de que al variar estos parámetros se lograra abarcar las posibles situaciones a las cuales se ve sometido el enlace.

Junto a los parámetros anteriormente mencionados se encuentra el tráfico, el cual está presente en la mayoría de las pruebas realizadas con el fin de ser una herramienta para poder analizar el enlace en el peor caso posible, y así lograr determinar el desempeño del mismo desde las mejores hasta las peores condiciones de trabajo.

Antes de entregar las conclusiones de las pruebas realizadas es importante anotar que las pruebas se realizaron en un ambiente con condiciones controladas en donde factores como el clima y la distancia entre las estaciones siempre fue constante y por lo tanto no hacen parte del conjunto de características que afectan negativamente al enlace, ni conforman fuentes de error.

Se puede observar que para un tamaño de paquete pequeño, el consumo de ancho de banda es mucho mayor que para un tamaño de paquete grande, lo cual no es de extrañarse pues el procesamiento que debe realizar el router (PC) es mucho mayor, dando como resultado mayor pérdida de paquetes.

En el caso en el que se transmiten ráfagas de paquetes por más tiempo, se tiene un ligero aumento en la pérdida de paquetes ($\approx$20paquetes/min). Lo cuál indica que el ancho de banda calculado aproximadamente en las primeras pruebas (cortas duraciones, 1 segundo), se ve un poco disminuido, pues se transmiten menos datos por unidad de tiempo.

Cuando se comparan los resultados de las pruebas con y sin tráfico se nota claramente cómo este factor influye negativamente en el enlace, a tal punto que su desempeño se puede ver disminuido hasta en un 54%.

La pérdida de paquetes en la transmisión se puede atribuir a que se haya superado en algunos casos la velocidad límite de transmisión del enlace (2,048 Mbps) con lo que se empiezan a descartar paquetes en el router (PC). Además se debe recordar que todo enrutador posee un buffer donde almacena provisionalmente los paquetes que debe procesar, y cuando ese buffer se encuentra lleno la acción que realiza el router es descartar paquetes. Las tarjetas Synclink no constituyen un cuello de botella, pues son capaces de soportar velocidades mayores a 10 Mbps y el enlace implementado opera a 2,048 Mbps.

Todas las reglas de filtrado fueron aplicadas sobre la interfaz de salida, lo que significa que antes de descartar o aceptar un paquete, ya ha sido tomada la dedición de enrutamiento en el PC. Esto significa que el firewall fue probado en el peor de los casos y aún así presentó un alto grado de rendimiento.

CONCLUSIONES

Se consiguió implementar satisfactoriamente un mecanismo capaz de soportar posibles aplicaciones software encaminadas al procesamiento de paquetes a través de un enlace WAN. Para realizar un enlace de este tipo, se utilizaron como herramientas hardware indispensables un par de tarjetas de comunicación serial (adaptadores WAN Synclink para Linux/PC), que fueron instalados convenientemente en computadores con sistema operativo Red Hat Linux versión 9.0. Como una posible aplicación en este campo, se plantea una herramienta de software orientada hacia el filtrado de paquetes (Firewall), cuya finalidad es mejorar el rendimiento del enlace WAN.

Basándose en el hecho de que las guías de instalación distribuidas por el fabricante de los adaptadores WAN utilizados requieren para su entendimiento de un nivel avanzado en el manejo del sistema operativo Linux, además de que se encuentran solo en inglés y no son suficientemente claras, se adoptó como fase inicial de este trabajo la creación de una documentación que describiera paso a paso el proceso completo de instalación y configuración básica de dichos dispositivos. Esta documentación fue elaborada en español y puede ser catalogada como una guía para el usuario, pues en ella se presenta detalladamente el procedimiento necesario para que una persona con conocimientos mínimos y poca experiencia en el sistema operativo Linux pueda lograr satisfactoriamente la puesta en marcha de los adaptadores. Esta documentación podrá servir para futuras aplicaciones que requieran el uso de los adaptadores y quedará a disposición del laboratorio de Redes de Datos de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

Se implementó un enlace WAN entre dos PCs funcionando como enrutadores, utilizando cada uno de ellos dos interfaces de red, una LAN y una WAN

(adaptadores Synclink). El enlace establecido fue probado (utilizando el protocolo HDLC) inicialmente mediante la realización de varias pruebas básicas de ping entre las estaciones y finalmente se realizaron varias transferencias de archivos utilizando para ello un servidor FTP.

Por medio de la instalación del Firewall se logra controlar el gran flujo de datos que circula a través de una WAN, que generalmente provienen de una LAN que se encuentra emitiendo gran cantidad de datos (muchos de ellos innecesarios). Debido a esto se hizo necesaria la aplicación de ciertos criterios de filtrado para restringir el acceso y por lo tanto mejorar el rendimiento de la WAN. Para realizar esto se escogió una herramienta de programación de módulos del Kernel llamada Netfilter Hooks, mediante la cual se implementó satisfactoriamente una herramienta que se encarga de realizar un eficiente filtrado de paquetes sobre la WAN.

En cuanto a la programación de las reglas de filtrado, se encuentra que si dichas reglas son aplicadas para los paquetes entrantes (al PC), se obtiene un mejor desempeño del sistema que si se aplicaran a los paquetes salientes, debido a que estos últimos han sido atendidos por el proceso de enrutamiento del equipo para seleccionar la interfaz de salida, causando una mayor demora en el procesamiento del Firewall y por ende una disminución en el rendimiento del mismo.

Se pudo comprobar que en varios casos, en los cuales el firewall trabajó con su máxima carga en cuanto a procesamiento de líneas de código (varias reglas de filtrado encadenadas), se presentó una considerable disminución en el rendimiento general del Router (PC) y por tanto del enlace. En condiciones normales de filtrado, o sea cuando se utilizan reglas de filtrado individualmente y sin tratar de exceder las capacidades del firewall, se comprobó que el firewall como tal no afecta notoriamente la transmisión de datos a través del enlace.

Finalmente cabe resaltar que este trabajo deja abierta la posibilidad de considerar y estudiar futuras aplicaciones encaminadas al desarrollo y mejoramiento de redes de datos, principalmente WAN, donde la transmisión a través de estas se realice de una forma más eficiente y segura.

Bibliografía

1. STALLING William, COMUNICACIONES Y REDES DE COMPUTADORES, Sexta Edición, Pearson Educación S.A., 2000. Libro de estudios general sobre redes, con descripción de protocolos y sistemas de referencia.

2. TANENBAUM Andrew S., REDES DE COMPUTADORAS, Tercera Edición, Prentice Hall Hispanoamericana S.A., 1997.

3. Página principal de Microgate ®, proveedor de los adaptadores. http://www.microgate.com/ . Se encuentra toda la información técnica acerca de los adaptadores, además contiene enlaces a todos los archivos propios del adaptador los cuales pueden ser modificados para su configuración.

4. Página principal de Cisco® http://www.cisco.com

5. GONZALEZ S, Néstor, COMUNICACIONES Y REDES DE PROCESAMIENTOS DE DATOS. McGraw-Hill, Bogotá. 1987.

6. Held, G; "INTERNETWORKING LAN AND WANS. CONCEPTS, TECHNIQUES AND METHODS". 2ª Edición; Wiley, 1998

7. Página principal de Rad Data communications www.rad.com.

8. The linux kernel module programming guide. Peter Jay Salzman y Ori Pomerantz.

9. MGEN User's and Reference Guide.htm se encuentran las principales características del generador de tráfico. Además se encuentran enlaces para adquirir el software libre.

10. How to use Netfilter Hooks.
http://uqconnect.net/~zzoklan/documents/netfilter.html

11. Página principal de la ITU-T se puede acceder a ella por medio del enlace
www.itu.int/home/

12. Página principal de la ISO se encuentra en www.iso.org

13. Página principal de la IETF se encuentra en www.ietf.org

14. Página principal de la EIA se encuentra en el enlace www.eia.org

15. ASMi-52 Installation and Operation Manual. Rad Data Comunications.
http://www.digitalhermit.com/linux/Kernel-Build-HOWTO.html

16. Biblioteca de Consulta Microsoft® Encarta® 2003. © 1993-2002 Microsoft Corporation. Reservados todos los derechos.

ANEXOS

ANEXO A. SOFTWARE DE FILTRADO DE PAQUETES

A.1 ESQUEMA BÁSICO DE UN MÓDULO DEL KERNEL PARA REALIZAR EL FILTRADO DE PAQUETES

El siguiente es un esquema básico del programa de filtrado de paquetes, sin aplicar ninguna regla de filtrado específica (está configurado para dejar pasar todos los paquetes).

```
#define __KERNEL__
#define MODULE
#include <linux/module.h>          //Necesitado por todos los módulos
#include <linux/kernel.h>          //Necesitado para las alertas del kernel
#include <linux/netfilter.h>       //Netfilter Hooks
#include <linux/netfilter_ipv4.h>
#include <linux/skbuff.h>
#include <linux/ip.h>              //Para la cabecera IP
#include <linux/netdevice.h>
#include <linux/udp.h>
#include <linux/tcp.h>


//////////////////////////////////////////////////////////////////////
/*DECLARACIÓN DE LAS ESTRUCTIRAS PARA LAS FUNCIONES PRINCIPALES*/
//////////////////////////////////////////////////////////////////////

static struct nf_hook_ops pkfilter_in;    //NF_IP_PRE_ROUTING
static struct nf_hook_ops pkfilter_out;    //NF_IP_POST_ROUTING


//////////////////////////////////////////////////////////////
/*DEFINICIÓN DE PARÁMETROS Y/O VARIABLES DE FILTRADO*/
//////////////////////////////////////////////////////////////

//FILTRADO POR INTERFAZ
static char *interface_LAN = "eth0";
static char *interface_WAN = "mgsl0";

//FILTRADO POR DIRECCIÓN IP
//static unsigned char *direccionIP_1 = "\xc0\xa8\x2d\x21";  //192.168.45.33 Sala de Redes
//static unsigned char *direccionIP_2 = "\xc0\xa8\x2d\x2a";  //192.168.45.42 Sala de Redes
//static unsigned char *direccionIP_3 = "\xc0\xa8\x2d\x2b";  //192.168.45.43 Sala de Redes
//Adicionar más si es necesario
```

```
//FILTRADO POR PUERTO DE CAPA DE TRANSPORTE
//unsigned char *port_1 = "\x00\x19"; //Puerto 25
//unsigned char *port_2="\x00\x??"; //Puerto ????  Messenger
//Adicionar más si es necesario


//////////////////////////////////////////////////
/*DEFINICIÓN DE LAS REGLAS, FUNCIÓN PRINCIPAL*/
//////////////////////////////////////////////////

//Función para el tráfico entrante
unsigned int hook_func_in(unsigned int hooknum, struct sk_buff **skb, const struct net_device *in,
                          const struct net_device *out, int (*okfn)(struct sk_buff*))

{
   if (strcmp(in->name, interface_WAN) == 0)
   {
        /*  Espacio para definir las reglas en la interfaz WAN para el tráfico entrante  */
        return NF_ACCEPT;
   }

   if (strcmp(in->name, interface_LAN) == 0)
   {
        /*  Espacio para definir las reglas en la interfaz LAN para el tráfico entrante  */
        return NF_ACCEPT;
   }

   else
   {
        /*  Espacio para definir las reglas en la interfaz Loopback para el tráfico entrante  */
        return NF_ACCEPT;
   }
}

//Función para el tráfico saliente
unsigned int hook_func_out(unsigned int hooknum, struct sk_buff **skb, const struct net_device
                           *in, const struct net_device *out, int (*okfn)(struct sk_buff*))
{
   if (strcmp(out->name, interface_WAN) == 0)
   {
        /*  Espacio para definir las reglas en la interfaz WAN para el tráfico saliente  */
        return NF_ACCEPT;
   }

   if (strcmp(out->name, interface_LAN) == 0)
   {
        /*  Espacio para definir las reglas en la interfaz LAN para el tráfico saliente  */
        return NF_ACCEPT;
   }
```

```
    else
    {
        /*  Espacio para definir las reglas en la interfaz Loopback para el tráfico saliente  */
        return NF_ACCEPT;
    }
}

//////////////////////////////////////////
/* RUTINA DE INICIALIZACIÓN DEL MODULO*/
//////////////////////////////////////////

int init_module(void)
{
    pkfilter_in.hook     = hook_func_in;     // Función principal para el trafico entrante
    pkfilter_in.hooknum  = NF_IP_PRE_ROUTING;      //First hook for IPv4
    pkfilter_in.pf       = PF_INET;
    pkfilter_in.priority = NF_IP_PRI_FIRST;   //Prioridad

    pkfilter_out.hook     = hook_func_out;    // Función principal para el trafico saliente
    pkfilter_out.hooknum  = NF_IP_POST_ROUTING;      //First hook for IPv4
    pkfilter_out.pf       = PF_INET;
    pkfilter_out.priority = NF_IP_PRI_FIRST;   //Prioridad

    nf_register_hook(&pkfilter_in);     //Cargar function para tráfico entrante
    nf_register_hook(&pkfilter_out);    //Cargar function para tráfico saliente

return 0;
}


//////////////////////////////////////////
/* RUTINA DE FINALIZACION DEL MODULO*/
//////////////////////////////////////////

void cleanup_module(void)
{
    nf_unregister_hook(&pkfilter_in);
    nf_unregister_hook(&pkfilter_out);
}
```

Para activar alguna regla[20] de filtrado, se deben adicionar en las partes donde se encuentra el comentario que dice “Espacio para definir las reglas”.

[20] Referirse al capítulo 7 para ver las posibles reglas de filtrado a implementarse.

A.2 PROGRAMA PARA COMPILAR, INSERTAR Y REMOVER EL FIREWALL

El siguiente es un programa realizado en lenguaje de programación del shell (bash), que se encarga de realizar el manejo correspondiente a la activación y desactivación del firewall, el nombre del programa es "Menu.sh[21]".

```
#!/bin/sh
#Programa principal para el manejo de filtrado de paquetes.

accion=$1

#Nombre del Filtro: Seleccione el archivo correspondiente al código en lenguaje C del filtro.
FILTRO1=pktfilter_v3b.c
FILTRO1_mod=pktfilter_v3b.o
FILTRO1_name=pktfilter_v3b

INSMOD=/sbin/insmod
LSMOD=/sbin/lsmod
RMMOD=/sbin/rmmod
RM=/bin/rm
LS=/bin/ls
CC=/usr/bin/gcc
flag=none

clear
echo -e "\nPrograma principal para el manejo de filtrado de paquetes "

if [ "$accion" = "c" ] || [ "$accion" = "ci" ] || [ "$accion" = "ic" ] || [ "$accion" = "cr" ] ||
  [ "$accion" = "rc" ] || [ "$accion" = "cri" ] || [ "$accion" = "rci" ]; then
        $RM -f $FILTRO1_mod
        echo -e "\nCompilando $FILTRO1..."
        $CC  -c -Wall $FILTRO1
        if ( $LS | grep -i "$FILTRO1_mod" ) then
                echo -e "\nMódulo para $FILTRO1 generado\n"
        else
                echo -e "\nMódulo para $FILTRO1 no pudo ser generado\n"
        fi
        flag=yep
fi

if [ "$accion" = "r" ] || [ "$accion" = "cr" ] || [ "$accion" = "rc" ] || [ "$accion" = "ri" ] ||
  [ "$accion" = "cri" ] || [ "$accion" = "rci" ]; then
        echo -e "\nRemoviendo $FILTRO1_name..."
```

[21] Desde ahora se hará referencia a el como el programa manejador

```
        $RMMOD $FILTRO1_name
        echo -e "\nMódulo para $FILTRO1_name removido\n"
        flag=yep
fi

if [ "$accion" = "i" ] || [ "$accion" = "ci" ] || [ "$accion" = "ic" ] || [ "$accion" = "ri" ] ||
  [ "$accion" = "cri" ] || [ "$accion" = "rci" ]; then
        echo -e "\nCargando mÃ³dulo...\n"
        if ( $LSMOD | grep -i "$FILTRO1_name" ) then
                echo -e "\nEl filtro $FILTRO1_name ya se encuentra en funcionamiento\n"
        else
                $INSMOD $FILTRO1_mod
                echo -e "\nEl filtro $FILTRO1_name ha sido activado\n"
        fi
        flag=yep
fi

if [ "$accion" = "names" ]; then
        echo -e "\nPresentado por:"
        echo -e "\n\n\tAndrés Augusto Jácome Lobo"
        echo -e "\n\tTatiana Inés Navas Gó mez\n\n"
        flag=yep
fi

if [ "$flag" = "none" ]; then
        echo -e "\n\n\t¡¡¡Escriba una opción válida!!!\n\n"
        echo -e "FORMA DE USO:"
        echo -e "./Menu.sh acción\n"
        echo -e "Los posibles valores para el parámetro acción son:"
        echo -e "c = compilar programa"
        echo -e "i = insertar módulo"
        echo -e "r = remover módulo"
        echo -e "Combinaciones aceptadas:  ci, ic, cr, rc, ri, cri, rci.\n"
fi
```

Este programa permite compilar un programa en lenguaje C para generar un módulo cargable (el módulo es la versión ejecutable del Filtro). Para ello, debe ejecutarse desde un terminal adicionando la opción c (compilar módulo). Para activar éste programa, se deben guardar en la misma ubicación tanto el manejador (Menu.sh) como el código en lenguaje C del Firewall (pktfilter_v3b.c). Para ello se recomienda crear una carpeta (se llamará Firewall) y en ella guardar ambos programas. Si el programa del firewall se encuentra sin errores, la compilación tendrá éxito. Desde un terminal se debe ejecutar de la siguiente manera:

[Firewall]# **./ Menu.sh c**

Además, el programa manejador permite insertar al kernel el firewall previamente compilado, para ello debe ejecutarse desde un terminal adicionando la opción i (insertar módulo). Desde un terminal se debe ejecutar de la siguiente manera:

[Firewall]# **./ Menu.sh i**

Asimismo está en capacidad de remover el firewall si se encuentra en ejecución, para ello debe ejecutarse desde un terminal adicionando la opción r (remover módulo). Desde un terminal se debe ejecutar de la siguiente manera:

[Firewall]# **./ Menu.sh r**

El programa presenta la opción de realizar varias labores con una sola ejecución, por ejemplo que compile y seguidamente instale el módulo, o que revise si ya se está ejecutando el firewall y lo remueva para insertar uno nuevo. Para ello existen unas combinaciones de opciones aceptadas que son: "ci" o "ic" para compilar e insertar, "cr" o "rc" para remover y compilar, "ri" para remover e insertar como una manera de reiniciarlo y "cri" o "rci" para compilar, remover e insertar[22].

[22] Estas opciones se crearon para el caso en el que se necesite modificar muy seguidamente el firewall.

ANEXO B. SISTEMA OPERATIVO LINUX

B.1 ELECCIÓN DEL SISTEMA OPERATIVO

Para el correcto desarrollo del proyecto es necesario contar con un soporte software que permita la implementación de mecanismos los cuales serán la plataforma de trabajo en el proceso del filtrado de paquetes. Una herramienta básica para lograr este objetivo es el sistema operativo.

El sistema operativo debe ser robusto para soportar todas aquellas aplicaciones que se deseen desarrollar con el fin de implementar un mecanismo para el filtrado de paquetes. Además es indispensable que el sistema operativo garantice una alta confiabilidad en los procesos de transferencia de datos, para descartar errores (propios del SO) cuando se realicen análisis de desempeño en las aplicaciones creadas.

En este contexto se presenta el sistema operativo LINUX como la alternativa adecuada para la realización del proyecto. Linux es un sistema operativo que provee al usuario de una interfaz para interactuar de una manera eficiente y segura con el software instalado en la máquina, proporcionando altas prestaciones con un bajo consumo de recursos.

Este sistema operativo se presenta en forma de varias distribuciones. Estas incluyen el kernel (Núcleo) y las aplicaciones (las cuales pueden ser creadas por el usuario o ser adquiridas por medio del software libre que facilita el proveedor del sistema operativo).

Muchas de estas distribuciones incluyen además un sistema de empaquetamiento de todos los programas incluidos al momento de la instalación del sistema operativo que facilitan la instalación y configuración del software.

Las distintas distribuciones no presentan grandes diferencias entre si, sin embargo, pueden presentar algunas variaciones en el mecanismo de instalación inicial, en el conjunto de aplicaciones inicial que se entregan al usuario.

B.2 ALGUNAS DISTRIBUCIONES

En la siguiente tabla se presentan características básicas acerca de las principales distribuciones de LINUX.

DISTRIBUCIÓN	DESCRIPCIÓN
RED HAT	Probablemente la mas popular en la actualidad. En el momento de la instalación cuenta con paquetes de grandes aplicaciones en el campo comercial y técnico. Además sus paquetes son de libre distribución lo que le confiere mayor libertad en el manejo que se le pueda dar a la utilización de los mismos.
EURELIEC	Distribución en español basada en Red Hat.
DEBIAN	Presenta avances en el modo de instalación, dado que se presenta la opción de realizar la instalación guiada y de una forma más flexible comparada con otras distribuciones. Solo incluye software estrictamente libre (GPL[23]).
S.U.S.E LINUX	Distribución más popular para Alemania.

Tabla 13. Algunas distribuciones de Linux

B.3 VENTAJAS DE LINUX COMO SISTEMA OPERATIVO

El sistema operativo Linux posee herramientas útiles al usuario presente en una estación de trabajo dado que presenta entornos gráficos (ventanas) y modo texto (consola), además de herramientas adicionales como aplicaciones de productividad. A continuación se resaltan las características técnicas de este sistema operativo.

[23] GPL de sus siglas en ingles *General Public License*

- Linux como multitarea: Permite ejecutar simultáneamente varios programas que comparten el tiempo de procesador. La multitarea en Linux es además muy robusta y no produce errores en las ejecuciones debido a la planificación de las mismas.

- Linux multiusuario: Dado que Linux es un sistema multitarea, permite que varios usuarios accedan a la computadora y ejecuten programas que compartan la CPU. Además la ejecución de los procesos de cada usuario, su memoria, ficheros etc. están protegidos de modo que cada uno de ellos pueda decidir quien accede a sus recursos.

- Linux como multiplataforma: usa el concepto de kernel como base fundamental en la implementación de aplicaciones y se ha adecuado de tal forma que pueda ser adoptado por multitud de plataformas, las cuales mejoran el rendimiento y desempeño del sistema operativo y de todas sus aplicaciones. Entre las plataformas que más se destacan se encuentran Alpha, Power PC, Pentium, Pentium Pro, Pentium II, ARM, MIPS, SPARC, entre otras, lo que hace que el kernel de Linux pueda ser usado en múltiples arquitecturas.

- Linux incorpora conectividad: Linux ha sido en algunos ámbitos "el sistema operativo de red", esto es debido a su buena disposición en aplicaciones como servidor y por las características de conectividad que incorpora, que le permiten manejar protocolos como TCP/IP, IPX, ftp, telnet, NFS, etc., las cuales facilitan la comunicación con otras plataformas.

- Linux es de distribución Gratuita: Linux es completamente gratis y puede ser adquirido por medio de descargas en Internet, al igual que la gran mayoría de aplicaciones que corren sobre el, además todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; por lo

que Linux constituye la alternativa más adecuada al ajustarse a cualquier presupuesto.

- Linux sistema operativo Seguro: Linux no sólo proporciona el sistema de protección entre procesos y entre archivos, también provee métodos de protección de red como firewall, sistemas de cifrado de datos, entre otros.

- Linux como sistema operativo estable: Linux no presenta fallas debidas a su arquitectura ni a su programación básica (Kernel). Esto se debe a que Linux aísla los procesos y su ejecución de manera que no alteren el procesamiento de la información y el funcionamiento propio del sistema operativo.

Para el proyecto se escogió el sistema operativo Linux en su distribución Red Hat 9. Esta opción permite la implementación de todas las aplicaciones programadas para este proyecto. Además Red Hat es compatible con todos los elementos hardware sugeridos para la realización del proyecto.

ANEXO C. HARDWARE RELACIONADO

Para cumplir con el objetivo de la implementación del enlace WAN se hace necesaria la conformación de una plataforma la cual pueda soportar todos los requerimientos propios del enlace. Para tal fin las herramientas hardware designadas para la realización del proyecto son: adaptadores WAN, conectores, cables y módems.

C.1 HARDWARE PRINCIPAL: ADAPTADORES WAN

Los adaptadores son la herramienta hardware más importante en la conformación del enlace, debido a que en estos adaptadores es donde se realiza toda la configuración necesaria para conseguir la interconexión entre las redes y así conformar el enlace. Estos adaptadores en conjunto con algunas herramientas software configuradas adecuadamente, están en la capacidad de ofrecer una nueva alternativa para realizar enlaces entre redes WAN sin contar con el uso de routers[24]. Dadas las características mencionadas anteriormente se presentan los adaptadores WAN SyncLink de Microgate® [3].

Figura 5. Tarjeta SyncLink WAN adapter for Linux / PC

Este adaptador de comunicación serial de alta velocidad incluye drivers para

[24] Ver Anexo G, Linux como router

Linux[25] (versión 2.00 en adelante). Forma parte de una familia de adaptadores seriales multiprotocolo para estaciones de trabajo con ranuras de expansión ISA, PCI, y PCMCIA. Respecto a su software interno se caracteriza por utilizar principalmente los protocolos HDLC[26] y PPP. Toda la programación interna de drivers del adaptador se realiza por medio de la interfaz tty, la cual permite tener el control de todas las aplicaciones y demás opciones que se presenten a lo largo de la configuración total del adaptador. Los parámetros básicos del adaptador se encuentran en la tabla 14.

PARÁMETRO	VALOR
Número de puertos serie	1 puerto
Interfaz serial	RS-232
Formato de datos Seriales	HDLC
Controlador	16C32 USC
Rata de datos	Mayor a 10 Mbps
Dirección I/O	Plug and Play Configurable
Temperatura	0- 50ºC
Humedad	0 a 95% Sin condensación
Altura	-200 a +10,000 Pies
Tipo de tarjeta	Tipo estándar PCI Short card.
Largo	6.875 Pulgadas
Ancho	4.2 Pulgadas
Peso	5.5 Onzas
Corriente optima	23 mA
Voltaje optimo	12 V
Conector	DB – 25 (Macho)

Tabla 14. Parámetros del adaptador SyncLink

El software incluido en el código fuente de cada adaptador está bajo licencia pública general (GPL[27]). La descripción más detallada de las herramientas software incluidas en el adaptador y que son utilizadas en el proyecto se encuentran relacionadas en el Anexo F.

[25] Ver Anexo I LINUX.

[26] Ver Capitulo 1.

[27] GPL de sus siglas en ingles *General Public License*

C.2 HARWARE COMPLEMENTARIO: MODEMS

Para lograr una mayor velocidad de transferencia en el enlace es necesario contar con un elemento hardware adicional al adaptador que compense las limitaciones de velocidad propias de la tarjeta WAN; ésta limitación se debe a la baja capacidad que presenta su reloj interno. Este modem debe ser compatible con las interfaces disponibles por el adaptador y además garantizar una velocidad de transferencia de datos superior a 1Mbps. Para cumplir con los requerimientos anteriormente mencionados se utilizan en el proyecto los módems ASMi-52 de RAD Data Communications[28]. En la tabla 15 se presentan sus principales características.

CARACTERÍSTICA	VALOR
ALTURA	43.8mm
ANCHO	240mm
PROFUNDIDAD	170mm
PESO	0.5Kg
MODO DE TRANSMISIÓN	Opera en full duplex sobre 2 líneas
RATA DE TRANSMISIÓN	Opera en el rango de 64 – 2.304 Kbps
INTERFAZ QUE SOPORTA	X.21, V.35 y E1
FUENTE DE AC	100 a 240 VAC
FUENTE DC	24 VDC o -48 VDC

Tabla 15. Características principales del modem ASMi-52 de RAD Data Communications

Figura 6. Modem ASMi – 52

[28] Referencia Bibliográfica [7]

C.3 HARDWARE ASOCIADO: CABLES

Otras herramientas hardware de gran importancia son los cables, los cuales cumplen la función de ser los conectores entre las estaciones y los demás elementos que conforman la red. Para entender la posición exacta donde los cables ocupan una función importante dentro del enlace se puede recurrir a la ayuda de la figura 7.

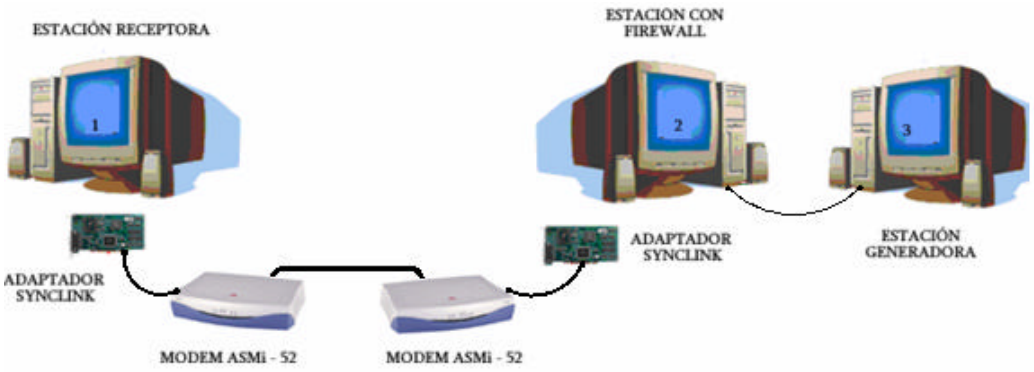

Figura 7. Esquema de la distribución de cables y conectores

De acuerdo con el esquema anterior se requieren tres clases de cables para realizar la interconexión entre los dispositivos del enlace. Como primer caso para la conexión entre los adaptadores SyncLink y los módems ASMi- 52 se requiere de un cable con conectores RS – 232 (Hembra) a V.35 (Macho). La forma del cable con sus conectores se presenta en la figura 8. Y sus correspondientes Pines de salida (PINOUT) se encuentran relacionados en la tabla 16.

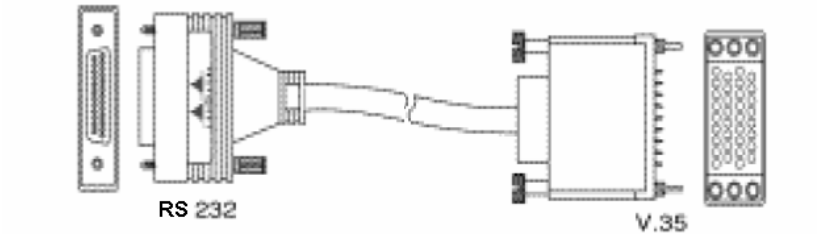

Figura 8: conexión entre Adaptador y modem

NOMBRE DE LA SEÑAL	EIA RS-232	V.35 (M)
Protección de Tierra	1	A
Señal de Tierra	7	B
TD(Transmitted Data+)	2	P
TD(Transmitted Data-)	14	S
RD (Recived Data+)	3	R
RD (Recived Data-)	16	T
RTS (Request to send)	4	C
CTS(Clear to Send)	5	D
DSR (Data Set Ready)	6	E
DTR (Data Terminal Ready)	20	H
DCD (Data Carrier Detect)	8	F
RI (Ring Indicador)	22	J
TTC (Transmit Clock+)	15	Y
TTC (Transmit Clock-)	13	AA
RC (Recive Clock +)	17	V
RC (Recive Clock -)	19	X
Auxiliar Clock +	24	U
Auxiliar Clock -	23	W

Tabla 16. Configuración de pines RS-232 a V.35

La gráfica correspondiente al cableado entre los dos conectores (RS-232 a V.35) se encuentra en la figura 9.

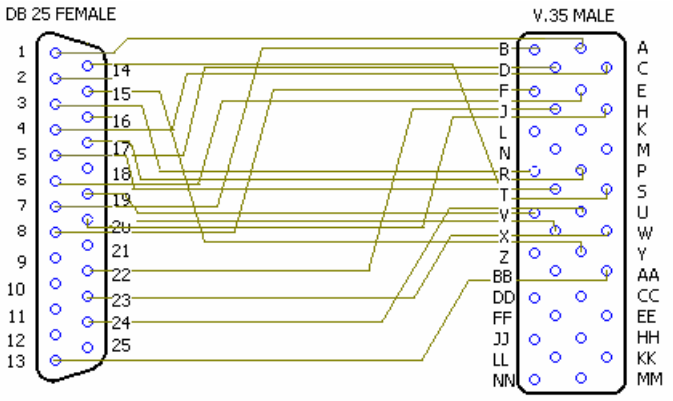

Figura 9. Conexión Rs 232- V.35

Continuando con la descripción de la gráfica, se requiere otra clase de cables para el enlace que son los utilizados para la conexión de los dos módems entre sí, al igual que para la conexión entre una estación del enlace WAN y una estación auxiliar; para este fin se utiliza un cable UTP categoría 5, cruzado.

Durante el desarrollo del proyecto fue primordial comprobar la conectividad entre las estaciones, par tal fin una de las primeras pruebas que se realizaron fue la conexión directa entre los adaptadores (Ver figura 10). Para esta prueba se debía prescindir de los módems para verificar el funcionamiento propio de las tarjetas. En este caso fue necesaria la utilización de un cable especial denominado Null modem. Este tipo de cable requiere de consideraciones especiales respecto a las señales de reloj manejadas en la transmisión de datos. Cuando se usa el Null modem uno de los adaptadores debe proveer la señal de reloj y permitir que por medio de su propio software se controle la velocidad del enlace. El adaptador SyncLink debe generar la señal de reloj y colocarla en el pin de reloj auxiliar de la interfaz serial. Posteriormente al conectar el null modem a cada uno de los adaptadores éste pone en contacto la salida de reloj anteriormente mencionada con las entradas de reloj que se encuentran en los dos adaptadores.

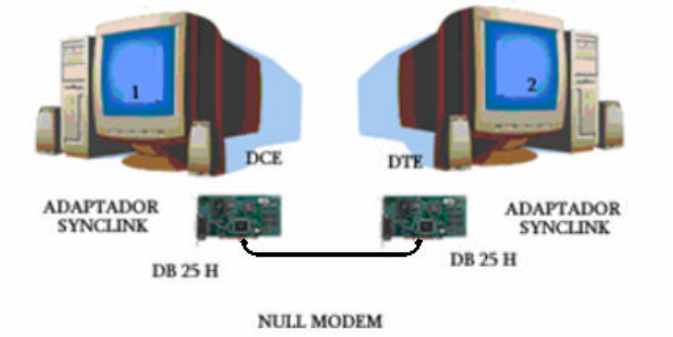

Figura 10. Conexión con NULL MODEM

En el caso particular del proyecto, el null modem presenta en cada uno de sus extremos un conector DB 25 (Hembra). En la tabla 17 se presenta la relación de los pines de salida (PINOUT) de cada uno de los conectores, y en la figura 11 se presenta el cableado entre ellos.

DB 25 (H)		DB 25 (H)	
NOMBRE DE SEÑAL	**PIN**	**NOMBRE DE SEÑAL**	**PIN**
DSR	6	**DTR**	20
DCD	8	**DTR**	20
---	---	**RTS, CTS**	4, 5
DTR	20	**DSR, DCD**	6,8
RTS, CTS	4,5	----	---
TxC, RxC, AUXClk	15,17,24	**TxC,RxC**	15,17
TxD	2	**RxD**	3
RxD	3	**TxD**	2
Signal Ground	7	**Signal Ground**	7

Tabla 17. Relación entre los conectores del Null Modem DB25 (H) a DB25 (H)

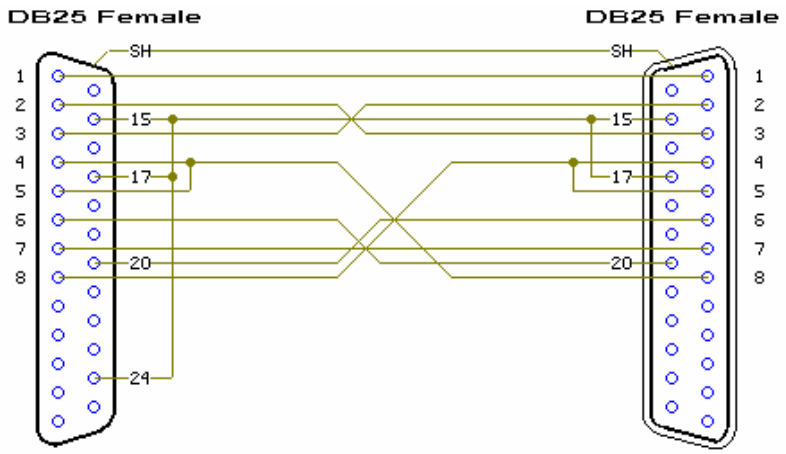

Figura 11. Cableado entre los pines del Null modem.

ANEXO D. INSTALACIÓN DEL SOFTWARE DE LA TARJETA SYNCLINK WAN ADAPTER EN UN SISTEMA OPERATIVO RED HAT LINUX

Un driver es un componente de software que permite controlar la comunicación entre un computador y alguno de sus periféricos (tarjeta de red, impresora, etc). Para el correcto funcionamiento de la tarjeta Synclink WAN adapter para Linux/PC, es necesario realizar la instalación de tres drivers cuyas funciones se describen a continuación.

Driver synclink.o

El driver usa la interfase del dispositivo tty estándar para proporcionar compatibilidad con las aplicaciones existentes. Se proporcionan mejoras en las comunicaciones síncronas y en la configuración de opciones.

Driver n_hdlc.o

Genera soporte para la disciplina de línea HDLC mediante un módulo cargable que permite usar el dispositivo de SyncLink para aplicaciones HDLC personalizadas. Este módulo altera el funcionamiento de las llamadas al sistema read/write estándar para un dispositivo tty.

Driver syncppp.o

Genera soporte para alternar entre el protocolo PPP y el protocolo Cisco HDLC, el cual implementa comunicaciones de red punto a punto, y es una alternativa a PPP en situaciones que requieren este protocolo. Éste driver está incluido en las distribuciones de Linux que son compatibles con las tarjetas Synclink WAN adapter de Microgate Corporation. El driver syncppp.o provee el soporte necesario para realizar la configuración IP de la tarjeta, y su respectiva activación.

Dependiendo de la aplicación en la cual se utilice la tarjeta, los drivers de Synclink pueden ser instalados de dos maneras diferentes, las cuales son descritas a continuación.

D.1 INSTALACIÓN DE LOS DRIVERS SIN SOPORTE PARA CONFIGURACIÓN IP

Es un método muy sencillo para construir los drivers y las utilidades de Synclink. Con este método se genera el soporte requerido por las aplicaciones de configuración de parámetros físicos de la tarjeta, además de las aplicaciones para realizar pruebas de transmisión en el enlace WAN.

Al finalizar este proceso de instalación, el sistema operativo Red Hat Linux estará en capacidad de utilizar los drivers synclink.o y n_hdlc.o. Mediante esta forma de instalación no se genera la posibilidad de manejar el driver syncppp.o, lo que indica que no se podrá cargar y configurar adecuadamente la interfaz de red de la tarjeta WAN.

D.1.1 Preparación del Kernel de Linux

La fuente del Kernel (Kernel Source) instalada con una distribución de RedHat requiere cierta preparación antes de ser usada para establecer los drivers de Synclink. Para esto es necesario que el sistema operativo Linux tenga incluidos los paquetes de desarrollo del kernel.

Con los siguientes pasos, se modificará la configuración del kernel, de manera que se puedan evitar incompatibilidades con los drivers y demás archivos que proporciona el fabricante. Esta configuración se realizó sobre Red Hat Linux

9.0 usando una versión de Kernel 2.4.20-8.

2. Ubicarse en el directorio base de la fuente del kernel. Para esta versión de kernel es "/usr/src/linux-2.4.20-8". Desde un terminal se debe utilizar el siguiente comando:

[root]# **cd /usr/src/linux-2.4.20-8**

Nota: En la sintaxis utilizada para especificar la línea de comandos, se especifica dentro del corchete la carpeta o ruta actual desde donde se ejecuta el comando, y después del símbolo # se encuentra la instrucción utilizada.

3. El siguiente paso es utilizar el comando 'make mrproper', lo cual borrará todos los archivos de configuración del sistema, excepto las fuentes (sources), incluyendo el archivo '.config' actual. Desde un terminal se debe utilizar el siguiente comando:

[linux-2.4.20-8]# **make mrproper**

4. Ahora el sistema operativo necesita un nuevo archivo de configuración, para esto se debe copiar el archivo de configuración que corresponda al equipo en donde se está realizando la configuración; éste archivo se encuentra en el subdirectorio de configuraciones, para esta versión de kernel es “/usr/src/linux-2.4.20-8/configs”, y se debe pegar en la carpeta base de la fuente del kernel (/usr/src/linux-2.4.20-8).

RedHat tiene archivos de configuración que han sido optimizados para diferentes tipos de equipos. Las opciones son ‘i586’ para los sistemas Intel® Pentium® y AMD K6™, ‘i686’ para los sistemas Intel® Pentium® II, Intel® Pentium® III e Intel® Pentium® 4, y ‘athlon’ para los sistemas

AMD Athlon™ y AMD Duron™.

Si el equipo tiene un procesador Intel® Pentium® 4, el archivo de configuración a elegir para esta versión de kernel es: “kernel-2.4.20-i686.config”. Éste archivo debe ser renombrado a “.config” solamente. Desde un terminal se deben utilizar los siguientes comandos:

```
[linux-2.4.20-8]# cd configs
[configs]# cp kernel-2.4.20-i686.config /usr/src/linux-2.4.20-8
[configs]# cd . .
[linux-2.4.20-8]# mv kernel-2.4.20-i686.config .config
```

5. A continuación es necesario editar el archivo 'Makefile' ubicado en el directorio base del kernel (para editar el archivo se puede utilizar un editor de texto como gedit, kwrite o vi).

Después, sólo si es necesario, se debe cambiar el valor de la extraversión el cual se encuentra en el comienzo del archivo para que concuerde con el valor de la extraversión del kernel en cuestión (La extraversión es el cuarto número que se encuentra en la versión completa del kernel de linux, por ejemplo linux-2.4.20-8 es versión 2, patchlevel 4, subnivel 20, y extraversión -8).

Desde un terminal se puede utilizar el siguiente comando:

```
[linux-2.4.20-8]# gedit Makefile
```

Nota: Es muy probable que en el archivo 'Makefile' aparezca la extraversión como -8custom y no -8. Se debe realizar el cambio correspondiente (Eliminar la palabra custom). Si la extraversión no concuerda con la del kernel, los drivers no serán cargados correctamente.

6. Desde la carpeta base del kernel (/usr/src/linux-2.4.20-8), se debe utilizar la instrucción 'make menuconfig', o si el sistema operativo tiene instalada la interfaz X-window será más sencillo utilizar la instrucción 'make xconfig', la cual presenta una interfaz gráfica más amigable para el usuario.

   Después de que aparezca la ventana de configuración de kernel, no se debe realizar ninguna modificación, solamente es necesario seleccionar 'Exit' y responder 'yes' para salvar la nueva configuración del kernel (si se utilizó menuconfig); si se utilizó x-config simplemente se debe hacer un click del mouse en 'save and exit'. Este paso crea los archivos necesarios basados en el nuevo fichero '.config'.

   Desde un terminal se puede utilizar uno de los siguientes comandos:

   [linux-2.4.20-8]# **make xconfig**

   [linux-2.4.20-8]# **make menuconfig**

   Además de estos dos comandos existe otro menos recomendable (modo texto), llamado 'make config', el cuál es más complejo y se presta más para cometer errores.

   Nota: Si se utiliza menuconfig es posible que en la ventana de comandos se produzca un error que dice: "Your display is too small to run menuconfig, it must be at least 19 lines by 80 columns"; esto solo significa que la pantalla del menú no puede visualizarse correctamente. En el menú 'Ver', se debe seleccionar alejar hasta cumplir con el requisito. Después se debe ejecutar de nuevo el comando.

7. Ahora es necesario crear ciertas dependencias, utilizando el comando 'make dep' (Solamente para versiones del kernel menores a 2.5)

   Desde un terminal se debe utilizar el siguiente comando:

   [linux-2.4.20-8]# **make dep**

Ahora el kernel está preparado para realizar la instalación de los drivers de Synclink.

D.1.2 Construcción de los drivers

En los siguientes pasos, se describe el procedimiento para construir los drivers de Synclink como procedimiento previo para su instalación. Esta configuración se realizó sobre Red Hat Linux 9.0 usando una versión de Kernel 2.4.20-8.

El primer paso consiste en adquirir el software del fabricante, éste viene incluido junto con la tarjeta, o se puede descargar vía web directamente con el fabricante en www.microgate.com.

Paso 1. Seleccionar el archivo que provee Microgate Corporation el cual contiene el software de Synclink, el archivo se llama “linuxwan.tar.gz” (formato comprimido), utilizando apropiadamente la instrucción ‘tar’ se debe descomprimir y extraer, se recomienda que este procedimiento se realice sobre la ubicación “/usr/src/”. Desde un terminal se deben utilizar los siguientes comandos:

```
[root]# cd /usr/src
[src]# tar xvfz linuxwan.tar.gz
```

Lo cual generará una carpeta llamada Synclink en la ubicación completa: '/usr/src/Synclink/'.

Nota: No es necesario extraer los archivos de Synclink estrictamente en esa ubicación, la instalación igualmente funciona desde cualquier ubicación, ésta solo es una sugerencia.

Paso 2. Antes de ejecutar los comandos correspondientes a la instalación de los drivers, es necesario cambiar el nombre a la carpeta que contiene los archivos de la fuente del kernel, la cual se denomina actualmente (para esta versión de kernel) "linux-2.4.20-8", y se encuentra en la ubicación '/usr/src/'. En esa misma ubicación se encuentra un acceso directo que se denomina "linux-2.4". Es necesario que la carpeta de la fuente del kernel se denomine "linux", para ello se puede utilizar otro acceso directo de 'linux-2.4.20-8', pero se debe renombrar a solo 'linux'. De esta manera se cumple con este requisito que exige los instaladores de los drivers de Synclink.

Paso 3. Dentro de la carpeta de distribución de Synclink (la que se creó con la instrucción de descomprimir en el paso 1), se debe usar la instrucción 'make driver'. Desde un terminal se deben utilizar los siguientes comandos:

```
[src]# cd Synclink
[Synclink]# make driver
```

Este procedimiento construye todos los drivers de Synclink, que serán usados con el kernel localizado en el directorio "/usr/src/linux". Los archivos binarios del driver están localizados en el directorio "/usr/src/Synclink/drivers/kernel-2.4".

D.1.3 Instalación de los drivers

A continuación se describe el procedimiento para instalar los drivers de Synclink. Esta configuración se realizó sobre Red Hat Linux 9.0 usando una versión de Kernel 2.4.20-8.

Dentro de la carpeta de distribución de Synclink, se debe utilizar ahora el siguiente comando:

[Synclink]# **make driver_install**

Para esta versión de kernel, este procedimiento copia automáticamente los archivos binarios a la ubicación 'lib/modules/2.4.20-8' y seguidamente ejecuta automáticamente el programa "depmod" para calcular las dependencias de los módulos.

Es posible resumir las instrucciones "make driver" y "make driver_install" en una sola instrucción "make driver driver_install". Para ello, desde un terminal se debe utilizar el siguiente comando:

[Synclink]# **make driver driver_install**

Después de esto los drivers de Synclink están listos para usarse.

Nota: En este paso es muy importante haber realizado correctamente los pasos correspondientes a la sección 3.1.1 (preparación del kernel de Linux). Específicamente el cambio de la extraversión, pues si no es correcta, surgirá un error inesperado en la ejecución del comando.

D.1.4 Instalación de las utilidades

Para poder utilizar las aplicaciones de configuración de parámetros físicos de la tarjeta, además de las aplicaciones para realizar pruebas de transmisión en el enlace WAN, es necesario realizar un paso previo, el cuál compila las utilidades en lenguaje C y crea los archivos ejecutables de las aplicaciones.

Para ello, desde un terminal se debe utilizar el siguiente comando:

[Synclink]# **make**

Después de esto las utilidades de Synclink, llamadas mgslutil y mgsltest, están listas para usarse.

D.2 INSTALACIÓN DE LOS DRIVERS CON SOPORTE PARA CONFIGURACIÓN IP

Es el método más complejo para construir los drivers de Synclink, en cuanto a que se requiere un mayor manejo del sistema operativo. Con este método se genera el soporte requerido por las aplicaciones de configuración de parámetros físicos de la tarjeta, además de las aplicaciones para realizar pruebas de transmisión en el enlace WAN.

Al finalizar este proceso de instalación, el sistema operativo Red Hat Linux estará en capacidad de utilizar los drivers synclink.o, n_hdlc.o y syncppp.o. Lo que indica que si es posible realizar la configuración de la interfaz de red para la tarjeta WAN (dirección IP, máscara de red, gateway, etc.).

D.2.1 Actualización de los drivers

Para utilizar este método, es necesario que los drivers de SyncLink formen parte de la versión de RedHat Linux que será utilizada. Si este es el caso, los drivers en la fuente del kernel podrán ser actualizados por unos más recientes en la carpeta de distribución de Synclink.

El primer paso consiste en adquirir el software del fabricante; éste viene incluido junto con la tarjeta, o se puede descargar vía web directamente con el fabricante en www.microgate.com.

Paso 1. Seleccionar el archivo que provee Microgate Corporation el cual contiene el software de Synclink. El archivo se llama "linuxwan.tar.gz" (formato comprimido), utilizando apropiadamente la instrucción 'tar' se debe descomprimir y extraer, se recomienda que este procedimiento se realice sobre la ubicación "/usr/src/". Desde un terminal se deben utilizar los siguientes comandos:

```
[root]# cd /usr/src
[src]# tar xvfz linuxwan.tar.gz
```

Lo cual generará una carpeta llamada Synclink en la ubicación completa: '/usr/src/Synclink/'.

Nota: No es necesario extraer los archivos de Synclink estrictamente en esa ubicación, la instalación igualmente funciona desde cualquier ubicación, ésta solo es una sugerencia.

Paso 2. Para actualizar los drivers de Synclink, el fabricante provee un archivo ejecutable llamado "INSTALL" que realiza automáticamente el proceso. Desde un terminal se deben utilizar los siguientes comandos:

```
[root]# cd /usr/src
[src]# cd Synclink
[Synclink]# ./INSTALL
```

El procedimiento anterior revisará las versiones de los drivers incluidos en su distribución de Linux, y preguntará para cada driver si se desea actualizar a una versión posterior, a lo que se debe responder Si. La actualización de los drivers no es un paso obligatorio en la instalación, pero es muy recomendado trabajar con unos que estén actualizados.

D.2.2 Instalación de los drivers

Una vez los drivers han sido actualizados, se debe proceder con la instalación de los mismos. Para ello es necesario configurar y compilar el Kernel de Linux, de tal manera que al instalar y ejecutar el nuevo kernel (recompilado), el sistema tenga incorporados en forma de módulos los drivers necesarios para el manejo de la tarjeta WAN.

Según esto, el siguiente paso en la instalación de los drivers es configurar y compilar el kernel, usando los procedimientos estándar presentados en el Anexo E[29] y teniendo en cuenta las opciones de configuración que se especifican en la sección 3.2.3 (Opciones de configuración del sistema para instalar los drivers).

Para ingresar al menú de configuración del sistema, se debe utilizar la instrucción 'make menuconfig', desde la carpeta base del kernel (Para esta

[29] Descripción del procedimiento estándar para configurar, compilar e instalar un kernel personalizado en Red Hat Linux.

versión de kernel es /usr/src/linux-2.4.20-8), o si el sistema operativo tiene instalada la interfaz X-window será más sencillo utilizar la instrucción 'make xconfig', la cual presenta una excelente interfaz.

Desde un terminal se puede utilizar uno de los siguientes comandos:

[linux-2.4.20-8]# **make xconfig**

[linux-2.4.20-8]# **make menuconfig**

Además de estos dos comandos existe otro menos recomendable (modo texto), llamado 'make config', el cuál es más complejo y se presta más para cometer errores.

Nota: Si se utiliza menuconfig es posible que en la ventana de comandos se produzca un error que dice: "Your display is too small to run menuconfig, it must be at least 19 lines by 80 columns". Esto solo significa que la pantalla del menú no puede visualizarse correctamente; En el menú 'Ver', se debe seleccionar alejar hasta cumplir con el requisito. Después se debe ejecutar de nuevo el comando.

D.2.3 Opciones de configuración del sistema para instalar los drivers

Si se utiliza alguna de las instrucciones recomendadas para ingresar al menú de configuración, entonces se podrá observar en la pantalla principal que todas las opciones se encuentran organizadas por categorías que se pueden señalar como principales, dentro de las cuales están las opciones para escoger, además de otras subdivisiones que se pueden catalogar como categorías secundarias. Las opciones que deben ser escogidas para activar cada uno de los drivers, en un kernel 2.4 o posterior son las siguientes:

1. Para habilitar el driver syncppp.o, en el menú de categorías principales seleccione 'Network Device Support' (Versión 2.4 de kernel).

    - Habilite (seleccionando “y”) la opción 'Network device support'
    - Habilite (seleccionando “y”) la opción 'PPP (point-to-point protocol) Support'
    - Habilite (seleccionando “y”) la opción 'PPP support for sync tty ports'

    Seleccione el Submenú 'Wan Interfaces'

    - Habilite (seleccionando “y”) la opción 'Wan interfaces support'
    - Habilite como módulo (seleccionando “m”) la opción 'SyncLink HDLC/SYNCPPP support'

2. Para habilitar los drivers synclink.o y n_hdlc.o. En el menú de categorías principales seleccione 'Character Devices'.

    - Habilite (seleccionando “y”) la opción ' Non-standard serial port support'
    - Habilite como módulo (seleccionando “m”) la opción 'Microgate SyncLink card support'
    - Habilite como módulo (seleccionando “m”) la opción 'HDLC line discipline support'

Después de que estas opciones han sido configuradas adecuadamente en la ventana de configuración de kernel, es necesario seleccionar 'Exit' y responder 'yes' (si se utilizó menuconfig) para guardar la nueva configuración del kernel; o si se utilizó x-config simplemente se debe hacer un click del mouse en 'save and exit'.

Las opciones requeridas para el soporte de Synclink dependen de la aplicación específica. Las opciones de SyncLink y de HDLC en el submenú ‘Character Devices’ son requeridas siempre. Las opciones de PPP en el submenú

'Networking device Support' son requeridas si se usa el demonio pppd para conexiones PPP. Las opciones de 'SyncLink SyncPPP support' son requeridas para usar alternativamente los drivers de PPP/Cisco HDLC. La localización exacta de cada opción puede variar para diferentes versiones del kernel.

D.2.4 Instalación de las utilidades

Para poder utilizar las aplicaciones de configuración de parámetros físicos de la tarjeta, además de las aplicaciones para realizar pruebas de transmisión en el enlace WAN, no es necesario realizar ningún paso adicional, pues el archivo ejecutable llamado “INSTALL”, el cuál debió ser utilizado anteriormente como lo indica el paso 2 de la sección 3.2.1 (Instalación de las utilidades), además de actualizar los drivers, realiza automáticamente la compilación de las utilidades que se encuentran en lenguaje C y para crear sus archivos ejecutables.

Después de esto las utilidades de Synclink, llamadas mgslutil y mgsltest, están listas para usarse.

ANEXO E. CONFIGURAR, COMPILAR E INSTALAR UN KERNEL DE LINUX

Este anexo proporciona una información muy detallada y un proceso paso a paso describiendo cómo configurar, compilar y posteriormente instalar un kernel Red Hat Linux personalizado. Este procedimiento ha sido llevado a cabo sobre un sistema operativo con la siguiente configuración: procesador AMD Athlon™ 1333 MHz, 80 GB de espacio en disco duro, 256 MB RAM, distribución Red Hat Linux 9.0 con kernel subyacente: 2.4.20-8.

1. Ubicarse en el directorio base de la fuente del kernel. Para esta versión de kernel es "/usr/src/linux-2.4". Desde un terminal se debe utilizar el siguiente comando:

   [root]# **cd /usr/src/linux-2.4.20-8**

   Nota: En la sintaxis utilizada para especificar la línea de comandos, se especifica dentro del corchete la carpeta o ruta actual desde donde se ejecuta el comando, y después del símbolo # se encuentra la instrucción utilizada.

2. Es altamente recomendable utilizar acontinuación el comando 'make mrproper', lo cual borrará todos los archivos de configuración del sistema, excepto las fuentes (sources), incluyendo el archivo '.config' actual. Desde un terminal se debe utilizar el siguiente comando:

   [linux-2.4.20-8]# **make mrproper**

3. Ahora el sistema operativo necesita un nuevo archivo de configuración, para esto se debe copiar el archivo de configuración que corresponda al

equipo en donde se está realizando la configuración; éste archivo se encuentra en el subdirectorio de configuraciones, para esta versión de kernel es "/usr/src/linux-2.4.20-8/configs", y se debe pegar en la carpeta base de la fuente del kernel (/usr/src/linux-2.4.20-8).

RedHat tiene archivos de configuración que han sido optimizados para diferentes tipos de equipos. Las opciones son 'i586' para los sistemas Intel® Pentium® y AMD K6™, 'i686' para los sistemas Intel® Pentium® II, Intel® Pentium® III e Intel® Pentium® 4, y 'athlon' para los sistemas AMD Athlon™ y AMD Duron™.

Si el equipo tiene un procesador Intel® Pentium® 4, el archivo de configuración a elegir para esta versión de kernel es: "kernel-2.4.20-i686.config". Éste archivo debe ser renombrado a ".config" solamente. Desde un terminal se deben utilizar los siguientes comandos:

```
[linux-2.4.20-8]# cd configs
[configs]# cp kernel-2.4.20-i686.config /usr/src/linux-
           2.4.20-8
[configs]# cd . .
[linux-2.4.20-8]# mv kernel-2.4.20-i686.config .config
```

4. A continuación es necesario editar el archivo 'Makefile' ubicado en el directorio base del kernel (para editar el archivo se puede utilizar un editor de texto como gedit, kwrite o vi). Es recomendable compilar el kernel con una extraversión diferente, con lo cual se previene que ocurra un cambio en alguno de los módulos existentes en su actual kernel, esto con el fin de mantener la configuración del kernel antiguo en caso de que ocurra algún error.

   Se debe cambiar el valor de la extraversión el cual se encuentra en las

primeras líneas del archivo. El valor que sea seleccionado para la extraversión será lo que diferencie al nuevo kernel del actual kernel (La extraversión es el cuarto número que se encuentra en la versión completa del kernel de linux, por ejemplo linux-2.4.20-8 es versión 2, patchlevel 4, subnivel 20, y extraversión -8).

Desde un terminal se puede utilizar el siguiente comando:

[linux-2.4]# **gedit Makefile**

Es muy probable que en el archivo 'Makefile' aparezca el nombre de la extraversión como -8custom o simplemente -8. Para el caso de este ejemplo de configuración, la extraversión del kernel actual es -8 y será cambiada a -8custom, pero puede ser cambiada en general a cualquier nombre.

5. El siguiente paso es realizar la configuración del sistema. Se debe utilizar la instrucción 'make menuconfig', desde la carpeta base del kernel (Para esta versión de kernel es /usr/src/linux-2.4.20-8), o si el sistema operativo tiene instalada la interfaz X-window será más sencillo utilizar la instrucción 'make xconfig', la cual presenta una interfaz gráfica más amigable para el usuario. Desde un terminal se puede utilizar uno de los siguientes comandos:

[linux-2.4]# **make xconfig**

[linux-2.4]# **make menuconfig**

Además de estos dos comandos existe otro menos recomendable (modo texto), llamado 'make config', el cuál es más complejo y se presta más para cometer errores.

Nota: Si se utiliza menuconfig es posible que en la ventana de comandos se produzca un error que dice: "Your display is too small to run menuconfig, it must be at least 19 lines by 80 columns". Esto solo significa que la pantalla del menú no puede visualizarse correctamente; En el menú 'Ver', se debe seleccionar alejar hasta cumplir con el requisito. Después se debe ejecutar de nuevo el comando.

6. Después de realizado el paso anterior, aparecerá la pantalla de configuración del sistema. Le ofrecerá varias opciones para obtener un Kernel Personalizado. Después de escoger las opciones necesarias se debe seleccionar 'Exit' y responder 'yes' (si se utilizó menuconfig) para salvar la nueva configuración del kernel; o si se utilizó x-config simplemente se debe hacer un click del mouse en 'save and exit'.

7. Ahora es necesario crear ciertas dependencias, utilizando el comando 'make dep' (Solamente es necesario para versiones del kernel menores a 2.5). Desde un terminal se debe utilizar el siguiente comando:

   [linux-2.4]# **make dep**

8. Después de creadas las dependencias, se debe ejecutar 'make bzImage' (este procedimiento que tarda un tiempo considerable en ejecutarse). Este paso construirá una imagen del kernel comprimida y lista para funcionar.

   Tras terminar este paso, un fichero llamado "bzImage" se creará en el directorio “/usr/src/linux-2.4/arch/i386/boot”. Si dicho archivo no se encuentra en la ubicación, es probable que se haya cometido un error, en ese caso es recomendable repetir todos los pasos desde el principio.

Desde un terminal se debe utilizar el siguiente comando:

[linux-2.4]# **make bzImage**

9. Después de que el archivo principal del kernel ha sido creado (bzImage), el siguiente paso en el proceso de compilación es crear los módulos cargables que han sido seleccionados en el menú de configuración del sistema (este es uno de los procedimientos que tarda más tiempo en ejecutarse). Desde un terminal se debe utilizar el siguiente comando:

[linux-2.4]# **make modules**

10. El siguiente paso es instalar los módulos que fueron creados en el paso anterior, para ello debe utilizarse la instrucción 'make modules_install'. Para llevar a cabo este procedimiento es necesario poseer privilegios de administrador (se requiere la contraseña de root). Los módulos serán copiados dentro de una carpeta cuyo nombre es la versión completa del kernel (La que aparece en el Makefile), cuya ubicación completa para esta versión de kernel, es "/lib/modules/2.4.20-8".

Desde un terminal se debe utilizar el siguiente comando:

[linux-2.4]# **make modules_install**

11. A continuación se debe crear una imagen del "initial RAMDisk" (o simplemente initrd). Crear una imagen del initrd es una forma de solucionar problemas de arranque, dados porque los drivers o módulos que son necesarios para cargar el sistema de archivos no pueden ser cargados puesto que éstos pertenecen al sistema de archivos.

Para crear el initrd se utiliza la instrucción 'mkinitrd'. Desde un terminal

se puede utilizar el siguiente comando:

[linux-2.4]# **mkinitrd /boot/initrd-2.4.20-8custom.img 2.4.20-8custom**

Con éste comando, un archivo llamado “initrd-2.4.20-8custom.img” será creado en la carpeta “/boot”, que es el lugar donde se encuentran los principales ficheros relativos al arranque del sistema operativo. Mediante mkinitrd se crean imágenes del sistema de archivos, las cuales son apropiadas para precargar los módulos que sean necesarios para cargar el sistema de archivos.

12. Para llevar a cabo la instalación del nuevo kernel, es necesario actualizar la información del gestor de arranque de Linux. La instalación de Red Hat posibilita la elección de un gestor de arranque, LILO o GRUB. Cada uno de estos tiene un archivo principal en donde se especifica el sistema operativo que será cargado, además presenta la posibilidad de seleccionar sistemas operativos diferentes, así como compilaciones distintas del kernel de Linux.

El primer paso para la instalación es copiar el archivo “System.map” que se encuentra ubicado en "/usr/src/linux-2.4.20-8", y pegarlo en la carpeta "/boot", pero es necesario renombrar el archivo para que concuerde con la versión del kernel recién compilado. Para esta versión de kernel el nombre del archivo será “System.map-2.4.20-8custom”

Desde un terminal se puede utilizar el siguiente comando:

[linux-2.4.20-8]# **cp System.map /boot/System.map-2.4.20-8custom**

13. El siguiente paso es copiar el archivo “bzImage” que se encuentra ubicado en "/usr/src/linux-2.4.20-8/arch/i386/boot/", y pegarlo en la carpeta "/boot", es necesario renombrar el archivo a “vmlinuz” y se le debe adicionar la versión del kernel recién compilado, para esta versión de kernel el nombre del archivo será “vmlinuz-2.4.20-8custom”

    Desde un terminal se debe utilizar el siguiente comando:

    [linux-2.4]# **cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.20-8custom**

14. El último paso en la instalación del nuevo kernel es modificar el archivo de configuración del gestor de arranque. Si el gestor de arranque instalado es Grub, entonces dicho archivo, llamado “grub.conf” o en algunas distribuciones llamado “menu.lst” aparecerá en la ubicación “/boot/grub/”. Por el contrario, si LILO se encuentra instalado, entonces el archivo es “lilo.conf” y su ubicación es “/etc”.

    Para editar adecuadamente el archivo de configuración del gestor de arranque (Se puede utilizar un editor de texto como gedit, kwrite o vi), se debe tener en cuenta las siguientes indicaciones dependiendo del gestor de arranque que se encuentra actualmente instalado.

Configuración de grub.conf:

Desde un terminal se deben utilizar los siguientes comandos:

[linux-2.4]# **cd /boot/grub**
[grub]# **gedit grub.conf**

El siguiente es un ejemplo de un archivo antes de ser modificado, el cual no incluye ninguna entrada para el kernel nuevo.

```
# Note that you do not have to rerun grub after making changes to #this file
#boot=/dev/hda
default=0
timeout=10
title Red Hat Linux (2.4.20-8)
        root (hd0,1)
        kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/
        initrd /boot/initrd-2.4.20-8.img
```

Para modificar el archivo con el fin de agregarle una entrada para el nuevo kernel, simplemente se deben copiar las cuatro líneas correspondientes al antiguo kernel de linux (cuatro últimas líneas del archivo), luego se deben situar al final del fichero y se deben cambiar para que concuerden con los nombres de los archivos del nuevo kernel. El nuevo archivo es el siguiente:

```
# Note that you do not have to rerun grub after making changes to #this file
#boot=/dev/hda
default=0
timeout=10
title Red Hat Linux (2.4.20-8)
        root (hd0,1)
        kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/
        initrd /boot/initrd-2.4.20-8.img
title Kernel de Prueba (2.4.20-8custom)
        root (hd0,1)
        kernel /boot/vmlinuz-2.4.20-8custom ro root=LABEL=/
        initrd /boot/initrd-2.4.20-8custom.img
```

La frase que acompaña a la instrucción “title” es solo el nombre con el que se distinguirá el sistema operativo. Se puede seleccionar cualquier nombre y no necesariamente debe llevar incluido la versión del kernel.

Cuando el archivo “grub.conf” esté actualizado y todos los archivos estén en su sitio, el siguiente paso es reiniciar el equipo y en la ventana correspondiente al gestor de arranque, se debe seleccionar el nombre del kernel nuevo. Si ocurre algún error y el kernel nuevo no arranca, queda la posibilidad de arrancar con el kernel antiguo. Para esto se debe seleccionar el nombre del kernel antiguo en la ventana correspondiente al gestor de arranque (repita con cuidado todos los pasos).

Configuración de lilo.conf:

Desde un terminal se deben utilizar los siguientes comandos:

```
[linux-2.4]# cd /etc
[etc]# gedit lilo.conf
```

El siguiente es un ejemplo de un archivo antes de ser modificado, el cual no incluye ninguna entrada para el kernel nuevo.

```
boot=/dev/hda
        map=/boot/map
        install=/boot/boot.b
        default=test-2.6.0
        keytable=/boot/us.klt
        lba32
        prompt
        timeout=50
        message=/boot/message
        menu-scheme=wb:bw:wb:bw
        image=/boot/vmlinuz
                label=linux
                root=/dev/hda3
                read-only
```

Para modificar el archivo con el fin de agregarle una entrada para el nuevo kernel, simplemente se deben copiar las cuatro líneas correspondientes al antiguo kernel de linux (cuatro últimas líneas del archivo), luego se deben situar al final del fichero y se deben cambiar para que concuerden con los nombres de los archivos del nuevo kernel. El nuevo archivo es el siguiente:

```
boot=/dev/hda
        map=/boot/map
        install=/boot/boot.b
        default=test-2.4.20-8custom
        keytable=/boot/us.klt
        lba32
        prompt
        timeout=50
        message=/boot/message
        menu-scheme=wb:bw:wb:bw
        image=/boot/vmlinuz-2.4.20-8
                label=linux (2.4.20-8)
                root=/dev/hda2
                read-only
        image=/boot/ vmlinuz-2.4.20-8custom
                label=test-2.4.20-8custom
                root=/dev/hda2
                read-only
```

La frase que acompaña a la instrucción “label=” es solo el nombre con el que se distinguirá el sistema operativo. Se puede seleccionar cualquier nombre y no necesariamente debe llevar incluido la versión del kernel.

Después de realizar la modificación de “lilo.conf”, es necesario ejecutar el comando “/sbin/lilo” para instalar el nuevo kernel. Desde un terminal se puede utilizar el siguiente comando:

[etc]# **/sbin/lilo**

El siguiente paso es reiniciar el equipo y en la ventana correspondiente al gestor de arranque, se debe seleccionar el nombre del kernel nuevo. Si ocurre algún error y el kernel nuevo no arranca, queda la posibilidad de arrancar con el kernel antiguo. Para esto se debe seleccionar el nombre del kernel antiguo en la ventana correspondiente al gestor de arranque (repita con cuidado todos los pasos).

ANEXO F. ACTIVACIÓN DEL ENLACE WAN

UTILIZACIÓN DE LOS DRIVERS

Un driver es un modulo cargable que puede ser dinámicamente activado, detenido y reconfigurado. Para el correcto funcionamiento de la tarjeta Synclink wan adapter para Linux/PC, el fabricante provee dos drivers llamados synclink.o y n_hdlc.o; mientras que un tercer driver está incluido con las distribuciones de linux que soportan las tarjetas, llamado syncppp.o.

Los drivers son cargados y descargados usando las utilidades estándar para el manejo de módulos del kernel. Las principales son modprobe (cargar módulos), insmod (insertar módulos), rmmod (remover módulos) y lsmod (lista de módulos activos). Refiérase a las páginas man de éstas utilidades para más información.

Activación de los drivers

Con el fin de facilitar la activación de los drivers que permiten el correcto funcionamiento de las tarjetas wan, el fabricante provee junto con el software de instalación, una aplicación denominada “load-drivers.sh”, el cual se encarga de cargar los drivers y configurarlos de manera adecuada para su correspondiente utilización.

Para utilizar ésta aplicación, basta con ejecutar el programa desde un terminal de la siguiente manera:

[Synclink]# **./load-drivers.sh [nombre del dispositivo]**

Donde el parámetro nombre del dispositivo está dado por una convención preestablecida por el fabricante, según la cual se especifica que las tarjetas deben designarse como "synclink" o como "/dev/ttySL[30][0-9]". El intervalo de cero a nueve [0-9] indica que pueden conectarse múltiples tarjetas simultáneamente en un mismo computador; de esta manera para un computador que posee una sola tarjeta wan de Synclink, el nombre del dispositivo sería "/dev/ttySL0" o simplemente "synclink". Para este caso, el comando para cargar los drivers queda de la siguiente manera:

[Synclink]# **./load-drivers.sh /dev/ttySL0**

Para simplificar éste comando de activación, es posible evitar el uso del nombre del dispositivo cambiando la línea del programa donde se encuentra el parámetro "LOAD_SYNCLINK_WAN = 0" por otra que diga "LOAD_SYNCLINK_WAN=1". Donde el valor cero (0) indica que el parámetro se encuentra deshabilitado, mientras que un valor de uno (1) quiere decir que el parámetro LOAD_SYNCLINK_WAN se encuentra habilitado, lo cual muestra que se cargarán los drivers de la tarjeta wan. Para realizar este procedimiento, es necesario editar el archivo 'load-drivers.sh' ubicado en la carpeta de distribución de Synclink (para editar el archivo se puede utilizar un editor de texto como gedit, kwrite o vi). Desde un terminal se puede utilizar el siguiente comando:

[Synclink]# **gedit load-drivers.sh**

Una vez halla sido modificado éste parámetro, el comando para cargar los drivers queda de la siguiente manera:

[30] El primer dispositivo sería mgsl0, el segundo mgsl1, etc.

[Synclink]# **./load-drivers.sh**

Para comprobar que los drivers se encuentran activados, se puede utilizar el comando “lsmod” desde un terminal de la siguiente manera:

[Synclink]# **lsmod**

Mediante este comando podemos ver, el estado (ocupado = 1, sin uso = 0) de los módulos del sistema y drivers que se encuentren activados. Si en el proceso de instalación de los drivers, se escogió la modalidad para soporte de configuración IP, entonces entre los procesos activos del sistema deben aparecer los nombres de los tres drivers (syncppp, synclink y n_hdlc). Por otra parte, si se escogió la modalidad sin soporte para configuración IP, entonces solamente deben aparecer los nombres de los drivers synclink y n_hdlc.

Desactivación de los drivers

Con el fin de facilitar la desactivación de los drivers de las tarjetas wan, el fabricante provee junto con el software de instalación, una aplicación denominada “unload-drivers.sh”, el cual se encarga de desactivar los drivers “Synclink.o” y “n_hdlc.o”. Para utilizar ésta aplicación, basta con ejecutar el programa desde un terminal de la siguiente manera:

[Synclink]# **./unload-drivers.sh**

Para realizar la desactivación del driver “syncppp.o”, debe hacerse con el comando “rmmod”, cuya función es la de desactivar módulos del sistema. Desde un terminal se debe utilizar el siguiente comando:

[Synclink]# **rmmod syncppp**

Nota: Se debe tener especial cuidado en el manejo de los drivers y en general de todos los módulos del sistema, pues es posible que se generen errores graves que desestabilizarían el sistema operativo. Por tal motivo es necesario tener en cuenta la sección referente a las Consideraciones sobre el manejo de las utilidades y los drivers descrito posteriormente.

DESCRIPCIÓN Y FUNCIONAMIENTO DE LAS UTILIDADES DE CONFIGURACIÓN Y CONECTIVIDAD

Dentro de las herramientas de software que provee el fabricante, se encuentran dos utilidades principales (mgslutil y mgsltest), y varias utilidades extra que se encuentran basadas en éstas. Cabe resaltar que para poder utilizar estas aplicaciones basta con que se tengan instalados los drivers synclink.o y n_hdlc.o, no es necesario instalar syncppp.o. A continuación se explica el funcionamiento y la forma de utilización de las mismas.

Mgslutil

Es una utilidad de línea de comandos, que permite al usuario ver y modificar las opciones y los parámetros de configuración de los adaptadores wan de Synclink. Antes de realizar la correspondiente activación de las tarjetas para efectuar una transmisión, es necesario utilizar mgslutil para configurar los parámetros internos de la tarjeta wan.

Ésta utilidad puede ser activada desde un terminal de la siguiente manera:

[Synclink]# **./mgslutil [nombre del dispositivo[31]] [options]**

Donde el nombre del dispositivo es por ejemplo “/dev/ttySL0”, y las opciones pueden ser consultadas utilizando el mismo comando de la siguiente manera:

[Synclink]# **./mgslutil**

Entre las opciones que se pueden configurar en los adaptadores wan, se encuentran opciones de codificación serial (NRZ, NRZI, NRZB, etc.), modos de encapsulamiento (HDLC, PPP), si la transmisión es síncrona o asíncrona, la velocidad del reloj de sincronía[32], etc.

En caso de que se desee adaptar un dispositivo externo que genere sincronía, como es el caso de un modem o un dispositivo CSU/DSU, es necesario fijar la velocidad del reloj de sincronía (data rate) a cero (0 = deshabilitado).

Mgsltest

Éste es un programa ejemplo que demuestra la utilidad de los adaptadores Synclink para aplicaciones personalizadas con HDLC. Las tramas HDLC son enviadas y recibidas presentando una estadística de la transmisión la cual dice que tan confiable fue.

Vale la pena aclarar que éste programa no ajusta ninguno de los parámetros de transmisión; El programa mgslutil debe ser ejecutado primero para ajustar los parámetros requeridos.

Puede ser usado en combinación con un dispositivo de loopback externo tal como un CSU/DSU en modo loopback o con el conector de loopback externo

[31] Ver la sección de activación de los drivers

[32] Esta velocidad va desde 0 bps hasta 115,2 Kbps

que viene junto con el adaptador, para verificar la correcta operación del adaptador.

Si el adaptador de Synclink se encuentra configurado para realizar loopback interno, entonces mgsltest envía y recibe paquetes a si mismo. Si no se encuentra configurado en el modo de loopback interno, entonces mgsltest (en modo maestro) envía una trama a un dispositivo externo a través del enlace wan y espera que en respuesta le envíen la misma trama de vuelta. El dispositivo externo puede ser otro adaptador Synclink corriendo el programa mgsltest en modo esclavo conectados mutuamente a través de un cable de NULL modem.

Ésta utilidad puede ser activada desde un terminal de la siguiente manera:

[Synclink]# **./mgsltest [nombre del dispositivo] [options]**

Donde el nombre del dispositivo es por ejemplo /dev/ttySL0, y las opciones pueden ser consultadas utilizando el mismo comando de la siguiente manera:

[Synclink]# **./mgsltest**

Entre las opciones que se pueden seleccionar se encuentran el tamaño y el número de tramas a enviar, además de la modalidad de transmisión (maestro, esclavo).

PUESTA EN FUNCIONAMIENTO DEL ENLACE WAN

Activación del Enlace Wan

Después de realizar la activación de los drivers, el siguiente paso a efectuar es

la activación y puesta en funcionamiento del enlace wan. Si se escogió la modalidad para soporte de configuración IP, es necesario verificar que la interfaz de red para la tarjeta wan ha sido creada. Para ello desde un terminal se debe introducir el siguiente comando:

[Synclink]# **ifconfig -a**

Al utilizar este comando, se deben visualizar una interfaz de red llamada mgsl0[33], que corresponde a la de la tarjeta wan. Respecto a esta interfaz vale la pena aclarar que pese a que fue creada, no se encuentra activada.

Para realizar la activación de la misma, es necesario definir su configuración IP. En un enlace punto a punto, es necesario definir varios parámetros como la dirección IP del Host local (ip address), el nombre de la interfaz a configurar (mgsl0), la máscara de red (netmask), y la dirección IP del Host remoto (Pointopoint). Para ello es necesario utilizar el comando "ifconfig" desde un terminal de la siguiente manera:

[Synclink]# **ifconfig [name] [dir_IP_local] netmask [mask] pointopoint [dir_IP_remota] up**

Donde:

- name = Nombre de la interfaz de red. Ej: mgsl0
- dir_IP_local = Dirección IP del Host local. Ej: 200.0.0.1
- mask = Máscara de red. Ej: 255.255.255.0
- dir_IP_remota = Dirección IP del Host remoto. Ej: 200.0.0.2

Configurando la red con los parámetros anteriores, queda:

[33] Este valor puede variar entre el rango de [0-9], dependiendo de cuantas tarjetas wan fueron instaladas en el mismo computador (mgsl0 – mgsl9)

ifconfig mgsl0 200.0.0.1 netmask 255.255.255.0 pointopoint 200.0.0.2 up

Utilidades adicionales: Start-cisco, Stop-cisco

Éstas dos utilidades tienen como función facilitar la activación y desactivación de un enlace wan que utilice Cisco HDLC como protocolo de encapsulamiento.

Start-cisco puede ser fácilmente editado y modificado para ajustarlo a las necesidades de la red. Es un programa que reúne todos los procedimientos necesarios para la activación del enlace. Primero se encarga de cargar los drivers, luego utiliza la herramienta "mgslutil" para configurar los parámetros del adaptador para la transmisión (protocolo cisco, data rate, tipo de interface, codificación, etc.) y finalmente utiliza el comando "ifconfig" para ajustar los parámetros de red (nombre de la interfaz de red, dirección IP, máscara de red, dirección IP del host destino). Ésta utilidad puede ser activada desde un terminal de la siguiente manera:

[Synclink]# **./start-cisco.sh [nombre del dispositivo[34]]**

Si no se especifica el nombre del dispositivo, se tomará por defecto "/dev/ttySL0", que corresponde a tener una sola tarjeta wan por computador.

Se deben configurar las tres líneas correspondientes a los parámetros de Red. Un ejemplo es el siguiente:

```
IPADDR=200.0.0.1
NETMASK=255.255.255.0
POINTOPOINT=200.0.0.2
```

[34] Ver la sección de activación de los drivers

Además es necesario verificar la línea que corresponde al reloj de transmisión "GENCLOCK=112500". Cuando el valor de Genclock es cero (0) significa que el dispositivo tomará la sincronización de un dispositivo externo (modem, CSU/DSU ú otra tarjeta WAN).

Stop-cisco no necesita ser configurado, simplemente debe ejecutarse y éste se encargará de deshabilitar la interfaz de red de la tarjeta wan de Synclink. Ésta utilidad puede ser activada desde un terminal de la siguiente manera:

[Synclink]# **./stop-cisco.sh [nombre del dispositivo*]**

Al igual que con Start-cisco, si no se especifica el nombre del dispositivo, se tomará por defecto "/dev/ttySL0", que corresponde a tener una sola tarjeta wan por computador.

CONSIDERACIONES SOBRE EL MANEJO DE LAS UTILIDADES Y LOS DRIVERS

1. En la documentación del fabricante se da la sugerencia de crear un archivo de configuración rápida de interfaz de red llamado ifcfg-mgsl0, pero éste paso genera problemas y errores que dificultan el manejo del enlace. Es recomendable utilizar el enlace como se describe a lo largo de las secciones anteriores.
2. Cuando se activa laguna aplicación que para su funcionamiento requiera de la utilización de los drivers, éstos se mantienen ocupados. Cuando no hay aplicaciones utilizando los drivers[35], entonces pueden ser descargados utilizando "unload-drivers.sh" o "rmmod", pero si el driver permanece en uso cuando trate de ser removido aparecerá un mensaje

[35] Con el comando lsmod, un cero (0) en el uso del driver indica inactivo

de error diciendo que el dispositivo está ocupado y el driver permanecerá cargado.

3. Al realizar la activación de la interfaz de red de la tarjeta wan de Synclink, se mantiene ocupado (Estado = 1) el driver “synclink.o”. Al momento de desactivarla se desocupa el driver.

4. Se debe tener especial cuidado en no tratar de utilizar aplicaciones tales como “mgslutil” y “mgsltest” mientras la interfaz de red de la tarjeta wan de Synclink se encuentre activa, pues éstas utilidades necesitan tener el driver “synclink.o” disponible para su utilización, pues se pueden generar errores graves del sistema.

5. Después de usar la utilidad “mgslutil” no se mantiene ocupado ningún driver.

6. Después de usar la utilidad “mgsltest” no se mantiene ocupado ningún driver, a menos que se interrumpa el proceso de comunicación antes de terminar. Si este es el caso los drivers se mantendrán ocupados y no podrán ser utilizadas ni la interfaz de red ni otras aplicaciones. Para desocupar el driver se recomienda cerrar la sesión de linux o reiniciar el sistema.

7. El script start-cisco.sh que provee el fabricante está incompleto en la parte que respecta a la utilización de ifconfig. Es necesario agregarle la opción Pointopoint (ver la página man de ifconfig).

ANEXO G. PC EN LINUX FUNCIONANDO COMO ROUTER

Un Router es un equipo que se encuentra conectado a diferentes redes, y por cada red a la que está conectado, dispondrá de un interfaz de red y su correspondiente IP. Además, el router se encarga de hacer IP-FORWARD de paquetes, es decir, si por una interfaz de red llega un paquete cuyo destino es un host conectado a otra red, y tenemos un camino (en la tabla de enrutamiento) hacia ese host a través de alguna otra interfaz, el router se encargará de retransmitir el paquete por esa interfaz.

Una de las posibilidades que ofrece Linux es la de poder actuar como enrutador, es decir que un computador estará en capacidad de intercambiar paquetes de datos entre sus interfaces y tomar decisiones de enrutamiento al respecto.

En los siguientes pasos, se explicará brevemente la manera de configurar un PC en Red Hat Linux para que funcione como un enrutador utilizando IPv4 como protocolo enrutado.

Es necesario activar el reenvío de paquetes entre interfaces, para ello se debe poner en "1" el archivo llamado "ip_forward" y que se encuentra en la ubicación "/proc/sys/net/ipv4/". Desde un terminal (cualquier ubicación) puede utilizarse el comando "echo" de la siguiente manera:

[root]# **echo 1 > /proc/sys/net/ipv4/ip_forward**

Este comando debe usarse cada vez que se encienda el computador, pues esa configuración no es permanente. Para hacer que el computador funcione permanentemente con IP_forwarding, entonces se debe adicionar

esa línea (echo 1 > /proc/sys/net/ipv4/ip_forward) al final del archivo "/etc/rc.d/rc.local".

Para observar la tabla de enrutamiento del computador, desde un terminal (cualquier ubicación) puede utilizarse el comando "ip route list" o "netstat", después se debe decidir que rutas no conoce el PC para que sean adicionadas de manera estática. Desde un terminal (cualquier ubicación), se pueden utilizar estos comandos de la siguiente manera:

[root]# **ip route list**

[root]# **netstat -nr**

Si es necesario, entonces adicione rutas estáticas y un gateway por defecto. Para adicionar una ruta por defecto para las redes desconocidas (default gateway), entonces desde un terminal (cualquier ubicación) utilice el siguiente comando:

[root]# **route add default gw [next-hop[36]]**

Para remover la ruta por defecto (default gateway), entonces desde un terminal (cualquier ubicación) utilice el siguiente comando:

[root]# **route delete default**

Para adicionar la ruta para una red, entonces desde un terminal (cualquier ubicación) utilice el siguiente comando:

[root]# **route add –net [net address[37]] netmask**

[36] Next-hop es la dirección IP del siguiente salto.

[mask[38]] gw [next-hop[4]]

Para remover la ruta para una red, entonces desde un terminal (cualquier ubicación) utilice el siguiente comando:

[root]# **route delete –net [net address] netmask [mask]**

Para adicionar la ruta por defecto para un host específico, entonces desde un terminal (cualquier ubicación) utilice el siguiente comando:

[root]# **route add [host address[39]] gw [next-hop[40]]**

Para remover la ruta para un host específico, entonces desde un terminal (cualquier ubicación) utilice el siguiente comando:

[root]# **route delete [host address]**

Para mayor información sobre como configurar el enrutamiento, refiérase a la página man del comando “route”.

---

[37] Net address es la dirección IP de la red que se desea adicionar a la tabla de rutas.

[38] Mask es la máscara de la red que se agregará.

[39] Host address es la dirección IP del Host específico que se desea agregar a la tabla de rutas.

[40] Next-hop es la dirección IP del siguiente salto.