# ADMINISTRACION
# DE REDES
# GNU/LINUX

*GUIA DE ESTUDIO HACIA UNA CAPACITACION SEGURA*

FUNDACION
**Código Libre Dominicano**

*Antonio Perpinan*

Fundación Código Libre Dominicano
http://www.codigolibre.org
e-mail: info@codigolibre.org

(1) Padre Pina No. 102, Zona Universitaria,
Santo Domingo
República Dominicana

(2) Calle Estrella Sadhalá No. 5
Jardines Metropolitanos,
Santiago
República Dominicana

Diagramación y Diseño de portada:
Nicaury Benítez Cortorreal

ISBN
88-99999-99-9

Depósito legal:
b8888.99

Impreso y encuadernado por
IMPRESOS GAMMA
Calle #12
Los Ríos
Santo Domingo-República Dominicana

Escrito Bajo la Licencia GFDL 2004

Impreso en República Dominicana
Printed in The Dominican Republic

## Nota de CopyLeft

Este documento se puede distribuir y modificar bajo los términos de la Licencia Libre de Documentación General del GNU - FDL.

© 2002-06 Antonio Perpiñan. Este manual es software libre; puede redistribuirlo y modificarlo bajo los términos de la licencia GNU de Documentacion Libre publicada por la Free Software Foundation; tanto en su versión 2 como (a su opción) en cualquier versión posterior. Este manual se distribuye con el ánimo de ayudar, pero sin garantía alguna; ni siquiera la implícita de ser comercializable o la de ser apto para un propósito en particular. Para más detalles, vea la Licencia Pública General de GNU (COPY-LEFT). Tiene a su disposición una copia de la Licencia Pública General de GNU (CopyLeft) en la distribución GNU/Linux que acompaña este Libro o en nuestra página web.

También puede obtenerla escribiendo a la Free Software Foundation, Inc., 59 Temple Place Suite 330, Boston, MA 02111-1307, USA.

En el ánimo de colaborar se ruega a quien utilice en total o en parte en cualquier material que de él derive, y que se respeten los tèrminos de la Licencia bajo los cuales el es distribuido.
Si modifica y mejora este documento, rogamos se lo
indique a sus autores originales, mediante
info@codigolibre.org.

# ADMINISTRACION DE REDES GNU/LINUX

*GUIA DE AUTO ESTUDIO HACIA UNA CAPACITACION SEGURA*

# INTRODUCCIÓN

## LOS PROPÓSITOS DEL CURSO

Los profesionales de la tecnología de la información (TI) son críticos hoy día para el ambiente de negocio. Adquirir las herramientas y conocimiento disponible en la tecnología de hoy es vital. GNU/Linux y el Código Libre y Abierto han colocado un nuevo estándar en lo que es desarrollo e implementación de aplicaciones nuevas y personalizables. GNU/Linux continúa ganando espacio de reconocimiento entre los profesionales y administradores del TI debido a su flexibilidad, estabilidad, y su poderosa funcionalidad. A medida que más empresas utilizan GNU/Linux, crece la necesidad de soporte y planificación sobre la integración de GNU/Linux en infraestructuras nuevas y/o existentes. El rol del administrador es guiar la implementación y desarrollo de soluciones basadas en GNU/Linux. Su éxito o derrota dependerán de su conocimiento y experiencia de esta fantástica arquitectura.

Este curso es un repaso comprensivo de las características y funcionalidad de GNU/Linux, orientada a preparar al estudiante con las herramientas necesaria para la certificación. Explicación detallada se provee de los conceptos claves, muchos conceptos y utilidades de GNU/Linux son idénticos sin importar la distribución específica siendo utilizada. Algunas características están disponibles en algunas distribuciones, y otras son añadidas durante la instalación. La naturaleza de GNU/Linux y el Software Open Source, es tal, que cambios al fuente y cambio a funcionalidad de cualquier componente debe ser incluido en la distribución específica . Los conceptos sublime de las capacidades de GNU/Linux se mantienen consistentes a través de cada distribución, kernel y cambio de Software.

Este curso ha sido desarrollado de acuerdo con los estándares de la industria de certificación de GNU/Linux. Los objetivo de la certificación de Sair Linux y GNU han sido elementos claves en el desarrollo de este material. La secuencia de los exámenes de certificación Sair/Linux provee la gama más amplia de los conceptos necesario para dominar GNU/Linux. Los objetivos de las certificaciones LPI y RHCE también son incluidos. El CD interactivo y la página Web con el curso contiene videos digital y pequeñas prácticas de de selección múltiples igual a los del examen. En el libro LA GUIA DEL ESTUDIANTE se provee una guía específica para la preparación de la certificación.

Este libro provee los conceptos y principios fundamentales necesarios para administrar un sistema GNU/Linux. Los conceptos y las tareas de administración pueden ser un poco amplios. Se le dará una explicación del rol del administrador, estructura y función detallada del kernel, y cubriremos tópicos administrativos claves del manejo de paquetes, procesos, espacio de disco, Backups y los usuarios así como las tareas programáticas, y los Logs/Registros del sistema. Este conjunto de herramientas te permitirán apropiadamente administrar un sistema GNU/Linux sea este de unos cuantos hasta miles de usuarios. Estos capítulos también te proveerán la información que necesitas para Certificarte.

Fundamentos de GNU/Linux proporciona una introducción a profundidad de los conceptos y de los principios que son necesarios para instalar un sistema GNU/Linux y desenvolverse en los ambientes de ventana del X y de la línea de comandos. Este manual da la dirección paso a paso para las distribuciones importantes de GNU/Linux y su instalación, incluyendo RedHat, Debian, Mandrake y Slackware. Se enfatizan los conceptos de instalación, las utilidades, y la funcionalidad de GNU/Linux común a todas las distribuciones y estas se explican en detalle adicional. Un principiante o un experto pueden aprender o repasar los conceptos de particionar discos y localizar los archivos de configuración, usando el shell y las consolas, crear los scripts, y editar archivos de texto que permanecen dominantes, sin importar la nuevas herramientas gráfica, para los ajustes de configuración. Este conjunto de tópicos permitirá que usted instale y configure correctamente un sistema GNU/Linux. En estos capítulos también se le provee la información necesaria para certificar sus habilidades en GNU/Linux

## METAS DEL CURSO

Este curso le proveerá con la información que necesitas para completar lo siguientes tópicos:

- Describir los componentes estructurales y distinguir entre una distribución de GNU/Linux y otra.
- Describir software de fuente abierta (Software Open Source) y entre GNU/GPL.
- Crear los disquetes de arranque de instalación.
- Instalar las principales distribuciones de GNU/Linux: RedHat (RPM), Debian (DPKG) y Slackware (tar.gz).
- Utilizar los ambientes de escritorio KDE y GNOME.
- Instalar y configurar XFree86.
- Localizar y utilizar la ayuda en línea.
- Configurar el hardware del sistema.
- El uso de fdisk o el cfdisk para crear, corregir, y suprimir particiones del disco.
- Utilizar el LILO/GRUB para manejar opciones para cargar el sistema.
- Arrancar el sistema, cambiar los runlevels, y cerrar o re-iniciar el sistema.
- Utilizar los disquetes de rescate para iniciar un sistema que se ha dañado.
- Describir el sistema de archivos jerárquico de GNU/Linux y el papel de los directorios y archivos claves en la organización del sistema. Trabajar con eficacia en la línea de comando de Linux usando comandos comunes del shell, streams, tuberías, filtros, y cambio de dirección.
- Usar scripts del shell para realizar tareas repetitivas rápidamente.
- Abrir, corregir, y almacenar documentos de texto usando el editor ‘vi’.
- Manejar los sistemas de impresión locales.
- Describir algunas aplicaciones comunes disponibles al usuario para sus tareas, tales como: navegar en Internet y acceso a E-mail, procesamiento de textos, presentaciones, hojas de cálculo, y manejo de gráficos.

## EJERCICIOS

Los ejercicios en este manual son diseñados para dar practicas reales en los ambientes de redes y aislados (stand-alone o networking) al usuario. Es altamente recomendado que usted complete todos los ejercicios en cada capítulo antes de continuar al próximo. Entendemos que en raros casos tal vez esto no sea conveniente cuando estudia fuera del taller. Si por alguna razón no puedes completar un ejercicio por circunstancias ajenas, debes planificar completarlo tan pronto sea posible.

Existirán ejercicios que no podrás completar por el limitante de equipo ó software. No permita que esto le impida completar los otros ejercicios que resten en el capítulo ó módulo.

## TOME NOTA

Los ejercicios en este libro fueron diseñados para ser ejecutados en un equipo de prueba y nunca deben ser llevados acabo en uno trabajando y donde se ejecuten aplicaciones importantes. Instalar GNU/Linux, reparticionar para instalar GNU/Linux, o practicando los ejercicios en una LAN u ordenador de trabajo puede causar problemas de configuración, lo cual puede conllevar a perdidas irreparable de data y dispositivos periféricos. Por favor siempre recuerde esta advertencia. Es preferible que dediques una estación de trabajo para practicar estos ejercicios. Instalar GNU/Linux en una situación dual-boot es una alternativa razonable, pero aún así conlleva ciertos riesgos.

## WEB y CD

Una parte muy clave de esta serie de auto-aprendizaje es el portal de soporte. Las lecciones que le indiquen visitar la página web o el CD-ROM que le acompaña, a menudo, es para ayuda con los conceptos que son mejor entendidos después de una descripción visual. Los segmentos video Digital proporcionan una ilustración gráfica acompañada por una narración de los instructores. Estas lecciones son ideales, ambos como introducciones para afinar conceptos y para ayudar el refuerzo.

## RECUERDE

Como herramienta de soporte les ofrecemos el CD interactivo, incluido en este libro, y nuestra página web http://www.abiertos.org y allí accesar hacia la sección Linux-Certificación, estos contienen exámenes de prueba de las diferentes certificaciones. Recreamos el escenario de las preguntas de selección múltiples, multi-selección y falso verdadero. Es muy importante que tome muchas horas de practicas antes de intentar pasar el examen de certificación que le corresponda ya sea LPI ó RHCE.

CAPITULO 1

*REDES DEL SISTEMA OPERATIVO GNU/LINUX*

# FUNDAMENTOS PROTOCOLO DE INTERNET (IP)


| TÓPICOS PRINCIPALES | No. |
|---|---|
| Objetivos | 68 |
| Preguntas Pre-Examen | 68 |
| Introducción | 69 |
| Componentes Comunes de Redes | 70 |
| Medios de Transmisión | 81 |
| Tipos de Transmisión | 89 |
| Capa de Enlace de Datos | 92 |
| Estándares de Comunicación | 110 |
| Direccionamiento de Internet | 132 |
| Interfases de Redes | 155 |
| El Futuro el IPv6 | 176 |
| Resumen | 190 |
| Preguntas Post-Examen | 191 |


## OBJETIVOS

Al completar este capítulo usted podrá:

- Describa el proceso y los elementos claves de la configuración básica de una red.
- Describa la tecnología básica que son el motor de las redes de Internet, Ethernet y Redes de Area.
- Liste y describa la implementación de las interfases más comunes de redes.

### PREGUNTAS PRE-EXAMEN

Las repuestas se encuentran en el Apéndice A.

1. Defina una red.
2. ¿Por qué usar una red?
3. Liste los niveles del modelo de cuatro capa del TCP/IP, luego liste 2 protocolos usados por cada uno.

# INTRODUCCIÓN

Una Red (Networking) se define como dos ó más computadores conectadas entre si para compartir data. El trabajo en red permite que la información sea distribuida fácil y rápidamente a través de un sistema de protocolo, cables, y hardware. Usted puede hoy día utilizar tecnología alternativas como lo son, infra-roja ó laser, para permitirle la comunicación entre dos ó más computadoras.

Las redes son extremadamente popular hoy en día ya que ellas permiten que los usuarios compartan su data en una manera eficiente. Ya es cosa del pasado que los usuarios tengan que colocar su información en disquetes o imprimirlo para poder físicamente transportar su data de un lugar a otro. No existe mejor método alguno que la redes para organizar y facilitar el trabajo de el trabajo en grupo..

Como su nombre lo indica que una red son dos ó más computadoras conectadas entre si vía un medio físico y protocolos de comunicación, una red puede ser un negocio pequeño que se desenvuelve en una pequeña area, la cual se denominan Redes de Area Local (Local Area Network) ó LAN. Una red también puede conectar varias LAN a través de una larga distancia ó una Red de Area Extendida. También son conocidas como WAN (Wide Area Network). Es posible extender una serie de WAN colocadas en diferentes partes del mundo interconectados como una sola red permitiendo que millones de usuarios compartan sus datos, así como es el Internet. Antes de poder comprender el Internet, internetworking, y el TCP/IP, usted está en la obligación de entender como funcionan las redes tradicionales.

El TCP/IP permite que computadores de diferentes marcas, modelos, sistemas operativos y capacidades (desde un mainframe hasta una PC de escritorio) se comuniquen. Desde su adopción en el 1983 por las grandes redes que conformaron el Internet, el TCP/IP ha superado con crece todas las expectativas que se tenía de el. Hoy día es el conjunto de protocolo de red más usado en el mundo y es el protocolo que hace posible el Internet, el WAN del mundo.

En este capítulo discutiremos la historia del Internet, los modelos que usa el Internet para comunicarse, la arquitectura del Internet, los protocolos comunes usados por en el Internet, y los protocolos seriales de enlace (serial link protocols). También discutiremos y analizaremos los documentos de Requisición de Comentarios ó RFC (Request for Comments RFC), documentos que definen y sirven de referencia para los protocolos de Internet.

## CONCEPTOS DE REDES/NETWORKING

Existen varios conceptos fundamentales que son básicos al analizar ó diseñar una red. El concepto central y fundamental de una red es que permita la comunicación y la compartición de recursos entre dos o más computadoras. Los detalles de como esta dinámica ocurre y los elementos que diferencia una red serán discutidos en esta sección:

- Redes Tradicionales
- Redes de Internet/Internetworking
- Ambiente de Redes
- Modelos de Red
- Categorías de Red
- Topologías de Redes
- Sistemas Operativos de Red
- Servicios Básicos de Red

**Redes Tradicionales**

Tradicionalmente cuando una compañía elige una red, esta trata de que la red que elige y luego use sea de un solo fabricante. Este tipo de elección creaba las redes homogéneas y centradas a una marca de fabricante en específico. La gran mayoría de organizaciones elegían vendedores como IBM, Novell, ó Banyan, para que le proveyeran soluciones de redes ya que redes compuestas por equipos de un sólo suplidor aseguran un mínimo de entrenamiento y capacitación para los empleados y sus profesionales de TI. El razonamiento era que utilizando el mismo tipo de redes creaba una situación lo más simple posible.

En el desarrollo de sus carrera puede ser que usted confronte mucho tipos de redes diferentes como las que hacemos aquí cita. Tiempo atrás el caso hubiese sido totalmente diferente y en sus labores lo más probable es que hubiese encontrado sólo un tipo de redes de compañía a compañía. Así es que hoy día debe tratar de familiarizarse con diferentes tipos de redes y también los protocolos en la diferentes situaciones que se le presenten, después de aprender este nuevo protocolo y topología, lo más probable que en el pasado hubiese sido todo lo que tenia que dominar para llevar acabo las tareas que necesita ejecutar.

**Redes de Internet/Internetworking**

Desde los años 90, han ocurrido cambios fundamentales sobre lo que redes concierne. Hoy en día muchos tipos diferentes de redes pueden trabajar interconectadas. Esta necesidad nace por causa de que organizaciones y sus divisiones necesitan estar permanentemente conectadas para compartir su información y recursos. La tarea de trabajar sistema diferentes en una manera heterogénea, como se trabaja interconectado a través del Internet, es conocido como internetworking.

Este tipo de trabajo en red representa un cambio dramático, ya que en el esquema tradicional de soluciones de redes, una organización sólo podía comunicarse consigo misma, pero para hacerlo con otras organizaciones, esta tenía que utilizar los métodos tradicionales, como lo fué tradicionalmente el correo. La motivación detrás de esta interconexión de las organizaciones y sus dependencia es causada por la necesidad de la transferencia continúa de información a través de extensas areas geográficas a muy alta velocidad. Hoy día será muy común que usted tenga que interconectar diferentes tipos de redes bajo una sola unidad lógica la cuáles pueden serán accesadas a través de la red pública, el Internet.

**Ambientes de Red**

Hoy en día existen dos principales ambiente de red en uso hoy: Redes de Area Local (LANs) y Redes de Area Amplia (WANs).

***Redes de Area Local (LANs)***

La definición más general de una red de área local (Local Area Network, LAN), es la de una red de comunicaciones utilizada por una sola organización a través de una distancia limitada, la cual permite a los usuarios compartir información y recursos como: espacio en disco duro, impresoras, CD-ROM, etc.

***Redes de Area Amplia (WANs)***

Es una red comúnmente compuesta por varias LANs interconectadas y se encuentran en una amplia área geográfica. Estas LAN's que componen la WAN se encuentran interconectadas por medio de líneas de teléfono, fibra óptica o por enlaces aéreos como satélites.

Entre las WAN's más grandes se encuentran: la ARPANET, que fué creada por la Secretaría de Defensa de los Estados Unidos y se convirtió en lo que es actualmente la WAN mundial: INTERNET, a la cual se conectan actualmente miles de redes universitarias, de gobierno, corporativas y de investigación.

**Tipos de Redes**

En sus orígenes las redes operaban con modelo centralizado, donde un mainframe suplía todos los recursos a computadoras denominadas estaciones de trabajo. Comúnmente esto limitaba sólo a redes de area amplia, que podían ser costeadas por grandes Universidades y compañías alto capital de inversión. A mediado de los años 80, la mayoría de estos negocios adoptaron el modelo cliente/servidor. En ves de usar servidores centrales del tipo mainframe, el modelo cliente/servidor utiliza un modelo modular que permite a pequeñas y medianas empresas utilizar estas soluciones de redes. Con la llegada del Internet, se pudo superar otra etapa de los modelos de red, que fué cuando estas pequeñas y medianas compañías pudieron utilizar económicamente modelos de redes de area amplia a través del uso de aplicaciones basado en la Web.

***Mainframes***

Sistemas Mainframe, también conocido computación centraliza, proveyeron la primera solución de rede practicas. Este método centralizado de emplear servidores centrales, o mainframe y terminales remotas. Normalmente estas terminales denominadas tontas o sin discos(diskless) solo pueden requerir información y no procesarla. El procesamiento de la información ocurre del lado del servidor.

Tradicionalmente obtener información desde el mainframe involucra un alto procesamiento por esta. Cuando un terminal pide información, esta requisición es enviada al mainframe. el mainframe procesa la requisición y obtiene la información deseada desde la base de datos o cualquier otra fuente. después que el procesamiento a culminado, el mainframe estructura y reforma la información al terminal. Veremos más adelante como el modelo cliente servidor difiere a este modelo.

El modelo de computación tipo mainframe tiene dos desventajas principales. La primera es que el mainframe efectúa todas las tareas de procesamiento. Las segunda es que los paquetes de petición y respuesta enviados entre el terminal y el mainframe ocupan mucho ancho de banda de la red. Estas dos desventajas en redes amplias y de mucho uso causan congestiones de red no aceptables.

Debido a la gran inversión que significan los mainframe en instituciones como, la banca, telefónicas, universidades, etc este modelo de mainframe es prevalente y lo será por mucho tiempo. Con el crecimiento de las tecnologías basada en la Web, las interfases basada en la web y otras tecnologías emergentes han reemplazado o por lo menos desplazado muchas de las plazas donde tradicionalmente se usaba un ambiente de mainframe y terminales.

***Modelo Cliente/Servidor***

La gran mayoría de aplicaciones en uso hoy día están basadas en el modelo cliente/servidor. En este modelo, la funciones se dividen entre los dos componentes de una aplicación. Basado en la concepción de que una parte proporciona el servicio (el servidor) y la otra usando el servicio (el cliente). Ejemplos de un servicio incluye acceso a una impresora, acceso a archivos almacenados en un sistema, ó acceso a una base de datos. Todos los servicios TCP/IP están basado en el modelo cliente/servidor. Normalmente, un servidor puede administrar las peticiones de multiple clientes.

El modelo cliente/servidor también es conocido como sistemas distribuidos, el cual reduce las congestiones de la red a través de una división de las tareas de procesamiento entre el cliente y el servidor. El servidor generalmente es más potente que que la computadora ejecutando el cliente y es responsable de almacenar y presentar la información.

En la mayoría de los ambientes de redes, existen varios Servicios disponibles. Es muy común que un computador sea un servidor para una aplicación y un cliente para otra. En una situación un computador puede ser un servidor de archivos, pero al momento de necesitar imprimir requiere de un servidor de impresión para que le lleve acabo esta función.

Otro buen ejemplo es sobre una WAN. Todos los que han navegado el World Wide Web (WWW) ha experimentado con aplicaciones cliente/servidor a través del suite de protocolo TCP/IP. Para accesar el WWW, el usuario ejecuta la aplicación cliente (en este caso el navegador Web, E.j.,., Mozilla, Firefox, Konqueror). Cuando el usuario desea accesar una página Web, este cliente contacta el servidor de páginas Web correcto sobre un enlace de red de área ampliada y nos despliega la página que el servidor Web nos envía.

En una sesión típica de navegación cuando un usuario sigue de un enlace a otro en un sitio web distinto, un cliente único contacto varios servidores. Al mismo tiempo, la máquina del usuario puede estar sirviendo también páginas Web con un servidor que están siendo accesadas al mismo tiempo por otros usuarios.

Una sesión típica Cliente/Servidor involucra el siguiente proceso:
1.- El usuario efectúa una petición de data.
2.- La computadora cliente traduce la petición en un formato que el servidor puede entender.
3.- El cliente envía la petición al servidor.
4.- El servidor procesa la petición, la cual puede involucrar comunicación con servidores remotos.
5- El servidor envía la repuesta al cliente.
6- El cliente envía la repuesta a su pantalla.

La diferencia clave entre este modelo de petición y repuesta y el que se presentó anteriormente usado por el mainframes es que el cliente procesa la mayoría de sus peticiones.

Además de compartir la responsabilidad de procesar las tareas, los beneficios de los clientes/servidores incluye una metodología modular del manejo del sistema. Debido a que el modelo cliente/servidor nos permite agregar nuevo componentes al sistemas, usted no esta limitado a una solo solución. Un tiempo atrás los ingenieros de sistemas tenían que elegir entre un sistema y el otro. Con la llegada de los Estándares abiertos como lo son el TCP/IP (Transmission Control Protocol/Internet Protocol), los sistemas heterogéneo pueden trabajar juntos más eficientemente. Por ejemplo, Servidores de UNiX y Windows NT que usan TCP/IP pueden trabajar juntos, permitiendole a los negocios escalar a soluciones de acuerdo con las demandas y necesidades del cliente. El modelo cliente/servidor es escalable porque este le proporciona la capacidad de ajustarse a nuevas demandas. El modelo Cliente/Servidor también concede a los usuarios más control sobre sus propios archivos.

Las organizaciones se encuentran cada día más y más en la necesidad de migrar hacia la arquitectura y aplicaciones basadas en el modelo cliente/servidor. Los sistemas de servidores y estaciones de trabajos basados en arquitecturas i386 (PC) brindan las mismas características encontradas en sistemas operativos de mainframe pero a un costo mucho más razonable para las pequeñas y medianas empresas. Se puede ir más lejo aún, existen hoy día miles de aplicaciones disponibles para estos sistemas operativos que se ejecutan sobre estas estaciones de trabajo y sus servidores.

La arquitectura Cliente/Servidor incorpora una flexibilidad considerable al distribuir los recursos en la red. Producto de la flexibilidad es la complejidad; la necesidad de entender los tipos de nodos de los clientes que son apropiados para las necesidades de su organización. Aún una selección más compleja es al escoger los nodos de servidores.

Las repuestas a muchas configuraciones del modelo cliente/servidor basados en:
• Los tipos de aplicaciones que caracterizan su ambiente de sistema
• Los usuarios y sus peticiones
• La arquitectura computacional y de redes

## Redes Basadas en la Web

Redes basada en la arquitectura Web, también conocida como sistemas colaborativos, utiliza tanto tecnologías mainframe y cliente/servidor. Con el uso de TCP/IP, tecnologías de trabajo basado en internet han permitido a las redes evolucionar a sistemas más y más decentralizados y distribuidos. En muchos de los casos, sistemas que se basan en la tecnologías Web representan la forma más radical del modelo computacional de tres capas. Redes altamente distribuidas, como son las extranets, las VPNs (Virtual Private Networks, Redes Privadas Virtuales), y Así sucesivamente, estas también dependen de navegadores de Internet para que sirvan de interfases de usuarios con mecanismo de servidores complejos.

El beneficio de este modelo es que puede combinar el poderío que representan los mainframe con la escabilidad del modelo cliente/servidor. Más aún, ya que las redes basadas en la tecnologías web dependen del uso de un navegador (browser), podemos presentar información sin la necesidad de programas especializados. Por esto es que redes de tecnología basada en la Web, proveen soluciones globalizadas que permiten a los usuarios obtener información y llevar acabo transacciones, independiente del sistema operativo que se encuentre y sin la necesidad de instalar programas adicionales. Las soluciones de comercio electrónico ó e-commerce, comprar carros, libros, etc, son un buen ejemplo de como se pueden llevar acabo transacciones seguras basadas en soluciones Web.

## Categorizando las Redes

Toda red consiste en tres mismos elementos básicos:

**Protocolos**

Las reglas de comunicación que todos los elementos de una red deben estar de acuerdo. En el próximo capítulo discutiremos los protocolos de redes a fondo.

**Medios Transmisión**

El método por el cual todos los elementos de la red se interconectan. Se discutirá más adelante.

**Servicios de Red**

Recursos como son las impresoras, que deben ser compartidas con todos los usuarios de la red. Se discutirá más adelante.

Al margen de estas similitudes, existen dos tipos básicos de redes: peer to peer (persona a persona) y basadas en servidores. Existe una tercera arquitectura, redes empresariales (enterprise), la cual combina las dos.

### *Redes Peer-to-Peer*

Una arquitectura de red peer-to-peer es aquella que no necesita de recursos dedicados, como es un servidor de archivos, además cualquier equipo (host) puede compartir sus recursos con otros sistemas en la red. En este tipo de redes, un mismo equipo puede funcionar como cliente y como servidor, ya sea haciendo peticiones de data ó sirviendola.

Por lo general, redes del tipo peer-to-peer tiende a resultar de menos inversión financieras y de menor requerimiento de conocimientos para quienes las operan que las redes del modelo cliente/servidor. Aunque estas redes son menos seguras y soportan menos usuarios (al rededor de diez ó menos es prudente), y experimentan más problemas con la administración de los sistemas de archivos debido a la falta de seguridad y control.

Ejemplos de productos de redes de PC del tipo peer-to-peer incluyen a:

- LANtastic
- Novell NetWare Lite
- Microsoft Windows for Workgroups (Para Grupos de Trabajos)
- Microsoft Windows 95/98

## Redes Cliente/Servidor (Server-Based Network)

Una red basada en servidores es una configuración de nodos de los cuáles algunos están dedicados a proveer recursos a los otros nodos en la red denominados hosts. Los nodos dedicados que ofertan sus recursos son denominados Servidores. Estos recursos incluyen impresoras, aplicaciones, y documentos.

Redes Basadas en Servidores ofrecen seguridad a los usuarios ya que se puede mantener un inventario en una base de datos central de los usuarios y que recursos ellos pueden tener acceso. En la otra mana, los servidores dedicados suelen a ser un poco más costosos y requieren un administrador de redes a menudo a tiempo completo.

**Algunos ejemplos de nodos de servidores comunes son**:
- Servidores de Impresión
- Servidores de Archivos
- Servidores de Correo
- Servidores de Páginas Web

Nodos Cliente pueden accesar estos recursos a través de la red. Ejemplos de redes del modelo cliente/servidor incluyen:
- Novell NetWare
- UNiX
- Microsoft Windows NT

**Redes Empresariales (Enterprise Network)**

Redes empresariales proveen conectividad entre todos los nodos de una organización, sin importar localización geográfica, y son capaces de ejecutar las aplicaciones de misión criticas. Estas redes incluyen ambos modelos que hemos discutido hasta el momento peer-to-peer y cliente-servidor.

Una red empresarial puede consistir de multiple pila de protocolos (stack) y arquitecturas de redes. Aquí le presentamos algunas de las características de las redes empresariales:
- Los sistemas de la red son capaces de trasladar paquetes de información desde una arquitectura a la otra (llamados gateways ó pasarelas).
- Los sistemas que soportan múltiples arquitecturas están presentes en la red (sistemas multi-protocolar).

Las Pasarelas ó Gateways, arquitecturas de redes y sistemas de multiprotocolos serán discutidos más adelante.

**Topologías de Redes**

Topologías son configuraciones básicas que el personal que manejan sistemas de información utilizan para cablear las redes. Ellas son el diseño básico de cualquier red. Las diferentes topologías usadas para conectar computadoras incluyen bus, star, ring, y híbrida.

**Topología Bus**

Topología Bus requiere que todas las computadoras ó nodos, estén conectadas a un mismo cable. Algunas computadoras envían data, esa data entonces es transmitida (broadcast) a todos los nodos en la red. El termino BUS se refiere a un AUTOBUS actual que debe detenerse en cada parada en su ruta. Sólo el nodo al cual el mensaje fué destinado leerá el mensaje; el resto de los nodos lo ignoran.

Oficinas y negocios pequeños a menudo usaban redes del tipo bus, ya hoy día con el abaratamiento de los equipos de redes como son HUBS, Switches, etc, este no es el caso. Si este tipo de red crece normalmente evolucionan a tipo start. Redes del tipo BUS requieren un terminadores en cada punta para asegurar que el tráfico no podruzca eco en toda la red.

Las redes Bus son relativamente simple, económicos, fácil de operar, y confiable. Además es muy eficiente en su uso de los cables. Una desventaja es que se dificulta diagnosticar problemas debido a que es difícil apartarlos; si un cable se rompe, la red entera puede ser afectada. Durante los momentos de mucho uso la red se tornará lenta.

**Topología Star (Estrella)**

Topologías del tipo estrella conectan nodos de redes a través de un dispositivo central, normalmente un hub (más adelante explicaremos que son los hubs). Como cada conexión de computador termina en el hub, este diseño reduce el riesgo de fracaso total de la red. Por ejemplo en el caso de la ruptura de un cable, sólo el nodo conectado directamente a este será afectado. El resto de la red funcionaría perfectamente bien.

Administradores de redes pueden resolver problemas mucho más fácil en redes del tipo estrella ya que los problemas son aislados automáticamente por el diseño. La red no es afectada cuando se desarrolla problema en un nodo. Expandir la red es simple y automática, la administración y el monitoreo puede ser centralizado. La desventaja es que como todo es centralizado a través del hub, si este falla toda la red se caerá.

**Topología Ring (Anillo)**

Topologías de tipo Ring no tienen un punto central de conexión. Estas redes tienen un cable que conecta un nodo al otro hasta que se forme un anilla. Cuando un nodo envía un mensaje, el mensaje es procesado por cada computador en el anillo. Si la computadora no es el nodo de destino, esta pasará el mensaje al próximo nodo hasta que el mensaje llegue a su destino correcto. Si por alguna razón el mensaje no es aceptado por ningún nodo, este dará la vuelta completa retornando a quien envió el mensaje originalmente.

Las redes de tipo Ring a menudo se conectan a través de un dispositivo central llamado Unidad de Acceso de Multiple Estaciones (Multistation Access Unit, MSAU), más adelante explicaremos que son estas unidades. Aislar los problemas en esta topología es difícil. Si se caé un nodo se caé todo la red.

Una ventaja de usar topología ring es que todas las computadoras tienen el mismo acceso a toda la data. Durante los período de uso pico, el rendimiento es igual para todos los nodos. Las redes tipo ring también ejecutan excelente en redes de alto tráfico de información. Una desventaja es que la expansión de la red o su reconfiguración afectará la red por completo.

**Redes Híbridas**

Puede ser que confronte grandes redes que combinen las topologías bus, star, y ring. Esta combinación permite expansión hasta en redes del modelo empresarial (enterprise networks). Por ejemplo puede ser que confronte una red star y una star bus. En una red star ring, dos ó más topologías star son conectadas usando un MSAU como un hub central.

En una red star bus, dos o más topologías star están conectadas utilizando un bus trunk. El bus trunk sirve como un backbone de la red.

Observe que cada red star contiene dos nodos y esta conectada por bus trunks. Esta topología es excelente para compañías grandes porque el backbone puede implementar media que soporte transferencias alta de data. Una ventaja de usar redes Híbridas es que las expansiones son relativamente simple. La redes no son normalmente afectadas si uno de los nodos falla. Pero si el Hub falla, las computadoras conectadas a ese hub no podrán comunicarse. Las conexione entre el hub que ha fallado y los otros también estarán fallida.

### Topología Mesh

Topología Mesh (Malla) interconectan dispositivos con multiple rutas para asegurar que exista redundancia. Todos los dispositivos están interconectados para así asegurar que la ruta más conveniente entre un dispositivo y otro es tomada en cada momento.

En una topología mesh, si una conexión es terminada ó interrumpida, otra conexión puede ser elegida para transferir la data a su nodo de destino. Una desventaja es que dispositivos (hardware) adicional tiene que ser empleado lo cual hace que que la topología mesh sea más costosa.

### Sistemas Operativos de Redes

Un Sistema Operativo de Redes (Network Operating System, NOS) administra recursos en una red. Sus funciones incluye administrar multiple usuarios en una red, proveyendo acceso a los archivos y servidores de impresión, y implementación de la seguridad de la red. GNU/Linux es uno de los mejores y más populares, al día de hoy, NOS.

Un NOS le permite a los clientes acceso remoto a los dispositivos, como si fuesen parte del equipo local del cliente. Ellos te permiten procesar peticiones desde los clientes y decidir si el cliente puede usar un recurso en particular.

Al igual que la relación cliente/servidor, una parte del NOS debe encontrarse ejecutando en el cliente, y la otra parte debe ejecutarse en el servidor. En una red peer-to-peer, cada cliente puede operar como ambos cliente ó servidor.

Muchos NOSs pueden operar entre ellos, esta característica es conocida como interoperabilidad. Esta característica facilita que las corporaciones pueda crear redes aunque sus clientes y servidores se encuentren ejecutando diferentes sistemas operativos. En la mayoría de los casos, software deberá ser instalado en los servidores y clientes para que exista esta interoperatividad.

### Servicios Básicos de Red

En un ambiente de redes hay disponible varios facilitadores básico:

**Distributed File System (Dfs)**

Sistemas de Archivos Distribuidos permiten a los clientes en una red accesar una jerarquía de archivos en un servidor central. Clientes y Servidores UNiX y GNU/Linux por lo general utilizan Network File System (NFS) para proveer Servicios de Dfs, mientras que MicroSoft utiliza el protocolo Server Message Block (SMB). Sistemas de Archivos Distribuidos son generalmente especificos a un sistema operativo de red cliente, aunque existen un número de utilitarios para accesar a NFS desde Windows y a SMB desde UNiX y GNU/Linux.

**File transfer**

Transferencia de Archivos permite que archivos individuales sean copiados desde un sistema a otro. Utilitarios de este proceso van desde mecanismos simple como el proceso único de Kermit al protocolo multitarea sofisticado proveído por el TCP/IP File Transfer Protocol (FTP).

**Remote printing**

Impresión Remota le permite al usuario en un siste imprimir un archivo en una impresora de otro sistema en la red. El otro equipo pueda o no estar ejecutando el mismo sistema operativo ó no. Esto sucede normalmente transparente al usuario y no requiere ninguna acción especializado por el usuario o el programa que este utilizando para generar el archivo a imprimir.

**Electronic mail**

Correo Electrónico permite que diferente usuario en sistemas diferentes se envíen mensajes entre ellos. Los sistemas de correo electrónicos son cada día más y más sofisticados; no es poco común poder enviar mensajes de voz y otros tipos de documentos multimedia con este método de comunicación.

**Telnet**

Protocolo de Terminales Virtuales proveé conexión entre máquinas y así permitiendo acceso remoto.

**Hypertext Transfer Protocol (HTTP)**

Protocolo de Transferencia de Hipertexto acepta peticiones de para documentos de del formato HTML (Hypertext Markup Language) y transfiere estos documentos al navegador (browser) remoto. Peticiones subsecuentes pueden incluir gráficos embebidos, fuentes tipográficas, javascripts y applets de java.

## HISTORIA DEL INTERNET

A través de su historia, donde se originó, quien la controla y quien la administra podemos poner en perspectiva esta tecnología tan reciente que a veces tiende a ser confusa para muchos de sus usuarios. En esta sección, discutiremos los siguientes tópicos:

- Interoperabilidad TCP/IP
- Internetworking y la Red Corporativas
- Evolución del Internet
- Puntos de accesos de la Red/NetworkAccess Points (NAPs)
- Las Autoridades Relacionadas con el Internet
- El RFC (Request for Comments)

### TCP/IP y la Interoperabilidad

El protocolo de red TCP/IP es el más usado hoy en día, esto no fué siempre el caso. Anteriormente, ya no muy reciente, redes de Novell usaban IPX/SPX (Internetwork Packet Exchange/ Sequenced Packet Exchange) como su protocolo por defecto de redes. Muchas redes de Novell aún continúan utilizando IPX/SPX y siguen siendo productivas. Muchas redes aún siguen usando protocolos que no son TCP/IP; estos protocolos son tan productivos como cualquier otro. Pero, si una red utiliza un protocol como por ejemplo NetBIOS (Extended User Inretface, NetBEUI) y otra usa el IPX/SPX, ellas no se podrán comunicar entre si. Estas redes deberán emplear dispositivos especiales, llamados gateways entre ellas para traducir los diferentes protocolos, pero una solución tal vez un poco más efectiva fuese adoptar el protocolo de red TCP/IP.

De echo estas redes no necesitan abandonar por completo los protocolos que ha estado usando tradicionalmente. Ellas pueden usar un protocolo internamente y usar TCP/IP como el protocolo que llevará acabo el transporte de la información entre su red y las otras.

El TCP/IP puede permitir que diferentes tipo de redes se comuniquen entre ellas. Usando sólo un router, el TCP/IP le permite a su LAN ó WAN existente intercomunicarse con otras.

El TCP/IP trabaja con muchas topologías diferentes. También puede funcionar en paralelo con otros protocolos a través de una sólo tarjeta de red (Network Interfase Card, NIC), ó encapsular otros protocolos dentro de los paquetes TCP/IP para interconectar dos redes que no usan TCP/IP vía una red de TCP/IP como es el Internet. Así esta actúa como un puente ideal que permite que redes LAN y WAN actuar como backbones para una empresa.

### Internetworking y la Red Corporativa

Hemos visto que el protocolo TCP/IP es ideal para internetworking (trabajo a través del Internet) ya que este permite que diferentes sistemas trabajan unidos. Este tipo de capacidad de trabajo de interplataformas significa que un equipo tradicional, digamos de IBM, puede comunicarse con soluciones cliente/servidores de hoy como lo son UNiX, Windows NT, Macintosh, y redes Novell. Los mainframe tradicionales anteriores pueden entonces trabajar entre ellas también. Como el protocolo TCP/IP es independiente de los fabricantes, este permite que trabajadores conectados vía el Internet se conecten desde cada sistema sin sacrificar la fortaleza de sus sistemas operativos ó su método de redes.

El protocolo TCP/IP ha sido atractivo para internetworking porque permite que las corporaciones y redes que sigan utilizando sus pasadas inversiones por mucho más tiempo. Por esto es que aunque el Internet y el

internetworking son revolucionarios, este protocolo presenta a los negocios una atractiva alternativa para que no tengan que eliminar flotillas de sistemas por anticuados. Con una buena planificación y trabajo, las compañías pueden hacer que sus equipos viejos se comuniquen con los equipos y sistemas nuevos y que trabaje con cualquier sistema sobre el Internet.

**Evolución del Internet**

El Internet se formó en el año 1968 cuando el Departamento de Defensa de los Estados Unidos creó la Agencia para Proyectos de Investigación Avanzada (Advanced Research Projects Agency, ARPA). Su primer red computacional global, fué el ARPANET (Red de la Agencia para Proyectos de Investigación Avanzada), que le permitió al gobierno y a los investigadores trabajar desde cualquier localidad que se pudiese conectar a la red. Las características de diseño del ARPANET contemplaba multiple clientes (hosts) y múltiples conexiones entre los clientes, lo cual reducía significativamente los chances de fracasos total de la red. No existía un hub central, lo cual hubiese creado un punto de vulnerabilidad; a diferencia, el control estaba distribuido en toda la red. Esta decentralización resulto en una red robusta y confiable que continuaría funcionando aunque muchos de sus clientes se cayesen y resultasen incapacitados.

Ya en el año 1971, el ARPANET consistía de 23 computadoras a todo lo ancho de los Estados Unidos, y al 1972, creció a 40. El año siguiente se extendió a Europa. Fué en este momento que esta colección de computadoras conectada en una red fué conocida como el Internet.

Al principio de los 80s, la versión del sistema operativo UNiX de Berkeley ya soportaba el protocolo TCP/IP, y en el 1981, TCP/IP pasó a ser el estándar oficial del Internet. El primero de Enero del 1983, el TCP/IP fué adoptado como el protocolo oficial del Internet. También en las década de los 80, el Departamento de Defensa asignó el proyecto ARPA a la Fundación Nacional de Ciencias (NSF). La NSF es una agencia independiente del gobierno de los Estados Unidos que promueve los avances de las ciencias y la ingeniería. La NSF incrementó el número de supercomputadores a cinco y agregó acceso a más redes, expandiendo sitios a negocios, universidades, al gobierno y instalaciones militares. Estos centros fueron conectados a través de líneas telefónicas de 56-Kbps que creaban redes regionales, con cada centro con un supercomputador sirviendo de hub para conexión de una región en particular.

El tráfico de red se incrementó significativamente. Un contrato fué obsequiado a la compañía Merit Network, Inc., lo cual permitió que las conexiones entre varias localidades operaran a velocidades de 1.5 Mbps.

En los años subsecuentes, se incluyeron más y más compañías privadas al Internet, hoy en día tecnologías existen para que se puedan adquirir velocidades de más de 2 Gbps y todos los días se establecen nuevos records de velocidad de transferencias. En la actualidad, tanto el hardware y los enlaces de comunicación que se requieren para interconectar el Internet son financiadas por una combinación del sector privado y del estado ó gobierno.

**Punto de Acceso de la Red (Network Access Points, NAP)**

Ya hemos dicho que el Internet es una serie de redes interconectadas. Un NAP es un punto donde se unen una red de alta velocidad y otra. Los tres puntos NAP claves en los Estados Unidos se encuentran en New York, Chicago, y San Francisco. Estos tres NAP son ejecutados por compañías telefónicas. Estos tres puntos de intercambio más el de Washington, D.C., son los cuatro originales en los Estados Unidos.

Estas redes de alta velocidad son las llamadas los backbones del Internet, ya que ellas proveen la conectividad esencial para el resto del Internet. Los Backbones pueden cubrir larga y cortas distancias, y redes más pequeñas típicamente se conectan a ellas.

La red backbone conectada por un NAP es una Red de Servicios Backbone de muy alta velocidad (very high-speed Backbone Network Service, vBNS). La mayoría de los NAP en uso al día de hoy son de velocidad en exceso de 1 Gbps y están diseñadas para reducir la congestión debido al uso incrementado continúo del Internet.

Por una mayor parte de la vida del Internet, agencias regionales y gubernamentales cargaban con la responsabilidad para proveer la conexión física. Empezando en el año 1995, los Proveedores de Servicios de Internet comerciales (Internet Service Providers, ISP) empezaron a financiar el Internet. Este grupo de ISP es llamado la Red Nacional de Investigación y Educación (National Research and Education Network, NREN). Esta usa los NAP para la conectividad.

**Segmentos/Segments**

Un segmento es un pedazo o parte de una estructura mucho más grande. En el Internet, un segmento puede ser parte del backbone que conecta San Francisco a Chicago. A escala menor, un segmento puede ser la conexión de la red en su oficina a su NAP. El término segmento también se refiere a una subred en una LAN.

**Las Autoridades Relacionadas con el Internet**

La autoridad de Internet descanza sobre el organismo llamado Sociedad del Internet (Internet Society, ISOC). ISOC es una organización de membrecía voluntaria de la cual sus objetivos es promover el intercambio de información a través del uso de la tecnología del Internet.

La ISOC escoge voluntarios quienes son responsables de la administración técnica y la dirección del Internet; estos voluntarios son denominados la Junta de Arquitectura del Internet (Internet Architecture Board, IAB). Usted puede visitar la sociedad del Internet en el sitio web www.isoc.org.

Otra organización voluntaria, llamada Equipo de Tareas de Ingeniería del Internet (Internet Engineering Task Force, IETF), se reúne regularmente para discutir problemas de operacionales y técnicos del Internet. Las recomendaciones son hechas vía grupos de trabajo dentro de la IETF y pueden ser enviados a la IAB para ser declarados como Estándares del Internet. El presidente de la IETF y los administradores de áreas conforman el Grupo Conductor de Ingenieros del Internet (Internet Engineering Steering Group, IESG).

Otra organización, llamada el equipo de Tareas de Investigación del Internet (Internet Research Task Force, IRTF), es responsable de investigación y el desarrollo de nuevas tecnologías aplicadas a la redes. El Grupo Conductor de Investigación del Internet (Internet Research Steering Group, IRSG) establece prioridades y coordina actividades de investigación.

**Petición de Comentarios (Request for Comments, RFC)**

Los RFC son documentos publicados de interés para la comunidad del Internet. Estos incluyen información detallada acerca de protocolos de Internet estandarizados y aquellos que se encuentran aún varios estados de su desarrollo. también incluyen documentos informativos relacionados con Estándares de protocolos, asignación de numerosa (E.j.,., números de puerto), requerimientos de host (E.j.,., Enlace de Data, Red, Transporte, y capas de Aplicación OSI), y requerimientos de enrutadores.

Los RFC son identificados por números. Mientras más alto es el número, es una indicación directa que es más actualizado el RFC. Asegurece de estar leyendo el más actualizado o reciente de los RFC durante sus investigaciones. El sitio web recomendado como referencia de los RFC es el sitio web www.rfc-editor.org/rfc.html. Observe que si un RFC ha sido actualizado, su listado nombrará su número de RFC reemplazo.

***Estados de los Protocolos***

Antes de que un protocolo se establezca como un estándar, este pasa por varios estados de madurez: experimental, propuesto, prueba, y estándar. Si un protocolo se torna obsoleto, se clasifica como histórico. Para que un protocolo puede progresar a través de estos pasos, el protocolo debe ser recomendado por el IESG del IETF.

Aquí le presentamos los diferentes estados de madurez de los RFC:

- **Experimental**
  Estos protocolos deben ser usados en situaciones de ejercicios y pruebas de investigación. Su intención no es para sistemas bajo operación normal sino para sistemas participando en la investigación del experimento.
- **Propuesto**
  Protocolos que están siendo considerados para una futura estandarización. Su puesta en prueba e investigación son promovidas (9999encouraged), es preferible que esas investigaciones sean llevadas acabo por más de un grupo. Estos protocolos serán revisados antes de pasar a la próxima etapa de su desarrollo.
- **Draft/Prueba**
  Estos son protocolos que están siendo seriamente considerados por el IESG para ser promocionados al estado de estándar del Internet. Pruebas son altamente 9999encouraged, los resultados de las pruebas son analizadas, y se requiere retroalimentación de información. Toda información pertinente a esta debe ser enviada al IESG. Muy a menudo cambios son efectuados en esta etapa de prueba; el protocolo dependiendo de estas propuesta es colocado en la etapa correspondiente a las propuesta.
- **Estándar**
  Estos protocolos son determinados por el IESG para convertirse oficialmente en protocolos Estándares del Internet. Los protocolos estándar se categorizan en dos tipos: esos que se aplican a todo el Internet y aquellos que sólo aplican a ciertas redes.

Otros estados de protocolos incluyen:

- **histórico**
  Estos son protocolos que han sido reemplazados por versiónes más recientes ó que nunca recibieron suficiente apoyo e interés para su desarrollo. Es muy difícil que un protocolo histórico se convierta en un protocolo estándar del Internet.
- **Informativo**
  Estos protocolos fueron desarrollados fuera del IETF/IESG (E.j.,. protocolos desarrollados por los fabricantes u otras organizaciones de estandarización). Estos protocolos son publicados para el beneficio de la comunidad del Internet.

***Estándares del Internet***

Un protocolo ó conjunto que ha sido elevado a estándar es indexado como estándar STD, como es el STD 5. Todos los protocolos, hasta los Estándares, son indexados como RFC porque los RFC nunca son eliminados, estos sólo cambian de estado del protocolo. Por ejemplo, IP es STD 7 Así como RFC 793. En algunos casos varios RFC pueden convertirse en un sólo STD. Por ejemplo, IP, ICMP, y IGMP son indexados todos en un sólo STD 5 aunque existen tres RFC diferentes: 791, 792, y 1112, respectivamente. estudiaremos estos protocolos en la próxima sección.

***RFC de Referencia***

Los siguientes son RFC de referencia que son importante y usted debe estar familiarizado con cada uno de ellos:

- **estándar Oficial del Protocolo de Internet, RFC 2700, STD 1**
  Este RFC lista el estándar actual del Internet Así como el estado actual de todos los RFC.
- **Números Asignados, RFC 1700**
  Lista el estado actual de los parámetros, como son los nombres y palabras claves, usadas en el Internet. Este incluye los números de protocolos asignados a los protocolos de Internet. Por ejemplo, el protocolo IP está representado por el número decimal 4. También se incluyen asignaciones bien reconocidas y registradas de puertos. Aprenderás sobre la asignación de números a través de este curso.
- **Requerimientos de Hosts de Internet, RFC 1122 y 1123**
  Este par de RFC define los requerimientos de software de host de Internet. Ellos definen los requerimientos únicos de protocolos dentro de la arquitectura del Internet y lista las características e implementación detalladas del protocolo (E.j.,., las especificaciones de los protocolos se identifican como ***must***, ***must not***, ***should***, ***should not***, y ***may***).

- **Requerimientos de Enrutadores de IP Versión 4, RFC 1812**
  El RFC 1812 define los requerimientos únicos de los enrutadores de IPv4 de Internet. Este actualiza el ahora histórico RFC 1716. Los requerimientos del enrutador incluyen la tecnología actual de enrutar.
- **Requerimientos de Enrutadores de IP Versión 6, RFC XXXXX**
  El RFC XXXXX define los requerimientos únicos de los enrutadores de IPv4 de Internet. Este actualiza el ahora histórico RFC 1716. Los requerimientos del enrutador incluyen la tecnología actual de enrutar.

## Ejercicios 1-1: Ubique los Documentos de los RFC

En este ejercicio, se le pide que ubique en el Internet los sitios de donde puede descargar los RFC. Usted debe investigar más de un sitio web ya que no todos los portales actualizan sus RFC con regular frecuencia. Soluciones para este ejercicio se proveen en el Apéndice B.

1. Desde el buscador www.google.com en un browser ubique tres sitios web que mantienen el listado de RFC. Liste los sitios por su URL como repuesta.

2.- Elija un sitio RFC y use el browser para ubicar los RFC 1250, 2068, y 1441. Liste los títulos de cada uno de ellos. Determine si el RFC es la última versión. Si no lo es, identifique cual es el más reciente. Usted deberá usar un editor de RFC que permita peticiones, como el que se encuentra en el sitio web www.rfc-editor.org/rfc.html. Elija buscar/Search desde el enlace índice de RFC que le permite buscar por número, titulo y autor.

| **RFC #** | **Titulo del RFC** | **RFC más actual.** |
|---|---|---|
| **RFC1250** | | |
| **RFC 2068** | | |
| **RFC 1441** | | |

3. Responda estas preguntas basada en la información en los RFC 2600 y 1700.
   a. ¿Es OSPFv2 un protocolo estándar del Internet? ¿Es RIP-2 un estándar?
   b. ¿En que año se constituyó POP3 como un protocolo estándar del Internet? ¿Y el TCP? ¿Y el IP?
   c. ¿Cúal es el STD para el protocolo de servicios NetBIOS?
   d. ¿Cúal es el rango de los números de puertos más reconocidos (well-known)?
   e. ¿Para que es el puerto 110 usado?
   f. ¿Cúal es el rango de los números de puertos registrados?
   g. ¿Para que es el puerto 533 usado?

## EL MODELO DE COMUNICACION DE DATA

En esta sección introducimos el modelo de referencia para la Interconexión de Sistemas Abiertos (Open Systems Interconnection, OSI) y el modelo de arquitectura de Internet. Estos modelos sirven para definir el mecanismo de todo protocolo de redes del TCP/IP usado para la comunicación de data.

En esta sección, discutiremos los siguientes tópicos:
- Protocolos de Comunicación
- Modelo de Referencia OSI
- Separación de Funcionalidad
- Arquitectura de Internet
- Paquetes/Packets

- Protocolos de Aplicación, Transporte, y Red
- Multiplexado/Demultiplexing
- Protocolos de Interfase Serial Especializada

**Protocolos de Comunicación**

Existe una necesidad de cierta forma de estandarización entre entidades que se comunican. Un protocolo es nada más que un conjunto de reglas y convenciones que describen como dos entidades se comunican una con la otra. Un ejemplo simple es un protocolo de comunicación es el lenguaje humano natural, digamos el español ó el ingles, estos permiten a humanos intercomunicarse uno con el otro y entender entre si que se esta diciendo. Es obvio que cuando dos computadores se intercomunican, el protocolo que se emplea deberá ser más definido, pero la idea básica es la misma.

Si dos personas no hablan un mismo idioma, ellos no podrán entenderse. Lo mismo pasa con dos computadoras que no entiende sus diferentes protocolos, para entenderse deberán emplear el mismo protocolo de comunicación para poder llevar acabo con éxito la comunicación.

***Capas de Protocolos (Layers)***

Los protocolos usados para intercomunicar computadores son bien complejos. Para orden administrar esta complejidad, los protocolos son divididos en una serie de capas, cada una de la cual es responsable de manejar cierto aspecto de la comunicación.

Los protocolos de cada capa se comunican con protocolos de su misma capa ó nivel en el sistema remoto. Realmente la data se transmite en orden decendiente a través de las diferentes capas en el sistema que envía ascendente en la capa del sistema que recibe. En cada capa el sistema que envía, coloca información de control al mensaje original. En el sistema receptor, esta información de control es interpretada por el protocolo homologo (peer) y esta información es removida del mensaje hasta Así llegar hasta el equipo receptor intacto como se envió el mensaje original.

Una colección de protocolos estructurado de esta manera aquí descripta es conocido como una pila (protocol stack).

***Información de Control***

Información de control agregada a un mensaje original al pasar entre protocolos puede ser comparada a un documento que ha sido colocada en una serie de sobres más grandes y más grande cada vez. Cada sobre tiene una dirección colocada en el para que Así el protocolo en ese mismo nivel en el host que recibe pueda entender para quien esta dirigida la data.

Al momento de recibir la data en el otro extremo, cada capa procede a destapar su sobre y pasa hacia arriba en la pila de protocolos el sobre que yace dentro del que esta dirigido a el, el determina a quien va dirigido por la dirección colocado en el sobre inmediatamente después del de el.

Cada capa en l apila no sabe nada de lo que contiene el sobre que el recibe desde arriba de la pila o por debajo de ella. Esto mantiene un diseño limpio y separado que permite a las capas trabajar independientemente.

**El Modelo de Referencia OSI**

El Modelo de Referencia OSI fué definido por la Organización Internacional para la estandarización (International Organization for Standarization, ISO). Fué introducida en el año 1983, el modelo OSI tiene tres funciones prácticas:

- Proveer a los desarrolladores los conceptos necesarios y universales para que puedan desarrollar y perfeccionar los protocolos.
- Explicar la estructura de trabajo usada para conectar sistemas heterogéneos. Más simple, permite a los clientes y servidores comunicarse aunque este usando aplicaciones y sistemas operativos distintos; todo lo que ellos necesitan es un protocolo común entre ellos como son por ejemplo el TCP/IP ó el IPX/SPX.
- Describir el proceso de la creación de los paquetes. Discutiremos más sobre la creación de los paquetes más adelante.

Redes son diseñadas y construidas usando el modelo OSI, en la misma manera que edificaciones son construidas usando planos. Por ejemplo, Novell NetWare, Windows NT, y GNU/Linux (UNiX) están todos basados en el modelo OSI. Este misma base arquitectónica de trabajo es que permite que estos sistemas se puedan intercomunicar.

Además, cuando quiera se discute de protocolos, como por ejemplo el IP y el IPX, normalmente estos son enlazados a su capa OSI. En este ejemplo, ambos protocolos se encuentran en la capa OSI de Red. El modelo OSI provee los conceptos y la nomenclatura que usted necesita para poder discutir lo que es la creación de paquetes y los protocolos de redes.

***Como se Comunican las Capas***

El modelo OSI describe la interacción entre las capas individuales Así como entre los hosts en la red. Un ejemplo de Cliente/Servidor será usado para explicar como trabaja típicamente el modelo OSI. La

| Capa | No. | Descripción |
|---|---|---|
| ***Aplicación*** | 7 | Esta es la capa presentada al usuario atraves de las aplicaciones disponibles en el ambiente OSI de su sistema operativo; esta capa soporta transferencia de archivos, administración de redes, entre otros servivios. |
| ***Presentación*** | 6 | Capa responsable de proveer transformaciones útiles sobre data para soportar interfases de aplicaciones estandarizadas y servivios de comunicación general. Por ejemplo, convierte texto de (ASCII) a Binario Extendido (EBCDIC). |
| ***Sesión*** | 5 | Establece, administra y termina conexiones (sesiones) entre las aplicaciones que colaboran entre ellas. Esta capa además agrega información de tráfico. |
| ***Transporte*** | 4 | Proveé transporte confiable y transparente entre dos puntos (Ej., los hosts de origen y destino). También proveé soporte de recuperación de errores y control de tráfico. Protocolos de estado de conxiones residen en esta capa. |
| ***Red*** | 3 | Esta capa es responsable de reenviar y enrutar los datagramas, (Protocolos sin estado de conexión residen en esta capa. |
| ***Enlace de Data*** | 2 | Proveé transferencia de data confiable atraves del enlace físico. Las tramas son transmitidas con la sincronización necesaria, control de error, y el control de flujol. Simplificando, esta prepara la información para que pueda ser enviada al cable físico. En las series de estandares IEEE 802 de de las LAN (un grupo de estandares populares de redes que estudiaremos más adelante), la capa de Enlace de Data está dividida en dos subcapas, el Control de Enlace Logico (Logical Link Control, LLC) y el Control de Acceso del Medio (Media Access Control, MAC). La capa LLC es responsable del control de error y flujo, y la capa de MAC es responsable de colocar la data en el cable. |
| ***Fisica*** | 1 | Capa que se concierne con lo relacionado con la transmisión de flujo de bits no estructurado sobre un enlace físico. Es responsable por las características mecanicas, electricas y procedural para establecer, mantener y deactivar el enlace físico. |

columna izquierda contiene las siete capas del modelo OSI que existen en el cliente. A la derecha la colum-

na con las siete capas que existe en el Servidor.

| | | |
|---|---|---|
| **Cliente** | | **Servidor** |
| **Aplicación** | **<<<------>>>** | **Aplicación** |
| **Presentación** | **<<<------>>>** | **Presentación** |
| **Sesión** | **<<<------>>>** | **Sesión** |
| **Transporte** | **<<<------>>>** | **Transporte** |
| **Red** | **<<<------>>>** | **Red** |
| **Enlace de Data** | **<<<------>>>** | **Enlace de Data** |
| **Física** | **<<<------>>>** | **Física** |

Si el cliente realiza una petición al servidor, la petición puede ser que empiece con un click del mouse por un usuario navegando en una página web (Capa de Aplicación). La petición viaja hacia abajo en la estructura del modelo OSI hasta que este llegue a la capa de Enlace de Data donde esta es entonces colocada en el alambre, al cable, ó cualquier otro medio que es usado en su red (la capa Física).

La petición del cliente viaja a través del cable hasta que esta llega al servidor. La capa de Enlace de Data toma la petición del cable (capa Física) y la envía hacia arriba en la estructura del modelo OSI. Cuando la petición llega a la capa de Aplicación del servidor, la petición es procesada. El servidor entonces retorna una repuesta al cliente, la cual puede ser una página Web nueva, usando el mismo método.

En la redes, la información como las peticiones del cliente y las respuestas del servidor son enviados a través de la red vía paquetes. Los paquetes sean discutidos más adelante en esta sección.

***Separación de Funcionalidad***

La motivación principal detrás del Modelo de Referencia OSI es la separación de funcionalidad entre las capas. Cada capa esta diseñada para ejecutar una función en específico y servir de interfase a otras capas en una manera bien definida. Los detalles internos de una capa no deben afectar la operación de las capas inferior o la superior a ella en la pila. De esta manera una capa puede ser removida completamente y reemplazada con otra. Siempre y cuando la interfase entre dos capas sea consistente, entonces este cambio no debe afectar la funcionalidad total de la red.

Esta forma idealista no siempre se puede lograr, pero la idea básica puede ser vista en un sin número de sitios. Una área de esta es la de direccionar el sistema. La capa de red superior debe operar independientemente del medio de red usado; por ejemplo, el uso de Ethernet ó Token Ring no debe afectar los protocolos de alto nivel. Esto nos deja dicho que los sistemas deben direccionar en una manera independiente del medio de conexión. Pero la capa de Enlace de Data debe poder direccionar sistemas, y el formato y la interpretación del direccionamiento dependerá del medio de conexión.

Así que la capa de Enlace de Data utiliza su propio formato de direccionar, el cual es entonces mapeado al nivel superior, direcciones independiente de la red como parte de la interfase entre las dos capas. Si un medio diferente de conexión es usado, entonces se requerirá de la interfase de ejecutar una forma diferente del mapeado de direcciones apropiado para la nueva capa de Enlace de Data. Pero, la operación del protocolo de la capa de Red no debe ser afectada por este cambio.

***Arquitectura de Internet***

Al igual que otros modelos de redes, la arquitectura de Internet divide los protocolos en capas. Cada capa es responsable de tareas de comunicación en específico. La arquitectura de Internet consiste de cuatro capas, cada una coincide con las capas del Modelo de Referencia OSI. Recuerde que existen varios

modelos de arquitectura de Internet, cada uno un poco diferente al otro. Aquí sólo trabajaremos con la versión de cuatro capas.

- **Capa de Aplicación**
- **Capa de Transporte**
- **Capa de Internet**
- **Capa de Acceso a la Red**

La siguiente tabla muestra la arquitectura de Internet y su equivalente en el modelo de capas OSI.

***La Capa de Acceso a la Red***

La capa de Acceso a la Red corresponde a las capas física y Enlace de Data del modelo OSI. Esta capa acepta paquetes de capas más alta y los transmite a través de la la red de conexión, manejando todos

| Modelo de Referencia OSI | Equivalente Modelo de Arquitectura de Internet |
|---|---|
| Aplicación | Aplicación |
| Presentación | |
| Sesión | Transporte |
| Transporte | |
| Red | Internet |
| Enlace de Data | Acceso a la Red |
| Física | |

los detalles del medio de las interfases de hardware con la red. Esta capa normalmente consiste de:

- Los manejadores de los dispositivos del sistema operativo
- La tarjeta de la interfase correspondiente
- Las conexiones físicas

Para las LAN basadas en tecnologías Ethernet, la data enviada por la media es referida como tramas de Ethernet, las cuales varían en tamaño desde 64 hasta 1,518 bytes (1,514 bytes sin la revisión de Redundancia Cíclica).

***La Capa Internet***

La Capa de Internet corresponde a la capa de Red del modelo OSI. Es responsable de direccionar y enrutar los paquetes en las redes TCP/IP. Un paquete recibido desde la capa de Transporte es encapsulado en un paquete IP. Basado en la información del host de destino, la capa de Internet usa un algoritmo de enrutamiento para determinar si envía el paquete localmente ó si necesita enviarlo a una pasarela (gateway) por defecto.

Los protocolos usados en la capa de Internet son:

- Protocolo de Internet (IP)
- Protocolo de Control de Mensajes de Internet (ICMP)
- Protocolo de Administración de Grupos de Internet (IGMP)
- Protocolo de Resolución de Direcciones (ARP)
- Protocolo de Resolución de Direcciones Inversas (RARP)

***Capa de Transporte***

La capa de Transporte del modelo de Arqitectura de Internet corresponde a las capas de Transporte y la de Sesión del modelo OSI. La capa de Transporte acepta la data desde la capa de Aplicación y le provee el flujo de información entre los dos hosts. Estos dos protocolos que siguen se encuentran en la capa

de Transporte:

- Protocolo de Control de Transmisión (TCP)
- Protocolo de Datagramas de Usuarios (UDP)

La capa de Transporte también divide la data recibida desde la capa de Aplicación en trozos de data más pequeños, llamados paquetes, antes de pasarla a la capa de Internet. La capa de Transporte también es conocido como la capa Host-a-Host, la capa Punta-a-Punta, ó la capa Fuente-a-Destino.

***La Capa de Aplicación***

La capa de Aplicación de la Arquitectura de Internet corresponde a las capas de Presentación y la de Aplicación del Modelo OSI. La capa de Aplicación interactúa con el protocolo de la capa de Transporte para enviar ó recibir data.

Los usuarios pueden invocar programas de aplicaciones, como es el protocolo de terminal remoto (Telnet), Protocolo de Transferencia de Archivos (FTP), Protocolo de Transferencia de Correo Simple (SMTP), ó Protocolo de Administración de Redes Simple (SNMP) para accesar los nodos en el Internet. La capa de Aplicación también suele ser referida de nombre la capa de los Procesos.

***Los Paquetes***

Un paquete es un pedazo fijo de información enviado a través de la red. En el momento que enviamos información a través de cualquier red, empezamos el proceso de creación de paquetes. Un paquete consiste de tres elementos: un cabezal (header), la data en si, y un finalizador (trailer). A menudo los paquetes también son referidos como datagram ó frame. Aunque este uso es aceptable la mayoría de las veces, un paquete es un nombre genérico usado para referirse a cualquier pedazo de información pasada por una red. Un datagram es un paquete en la capa de Red del modelo OSI. Un frame (trama) es un paquete en la capa de Enlace de Data . Aunque esta diferencia existe ellos son intercambiados sinónimamente.

El header contiene información variada, como es información referente a direccionamiento ó señales de alerta a la computadora entrante. El paquete contiene la data original, como por ejemplo un correo. El trailer normalmente contiene información que valida el paquete. Por ejemplo, puede contener la información Revisado de Redundancia Cíclica (CRC).

***Cyclic Redundancy Check (CRC)***

Un CRC es un calculo matemático que le permite al computador que recibe validar si un paquete es valido. Cuando un host envía un paquete, este calcula un CRC, entonces procede a agregar esta información al trailer. Cuando el host que recibe lee el paquete, este genera su propio CRC, entonces lo compara con el CRC que esta almacenado en el trailer. Si igualan, el paquete no esta dañado u el host que recibe procesa el paquete. Si los CRC no igualan, el host recibiendo lo descarta por completo.

***Creación de Paquetes- Agregando los Headers***

La creación del paquete empieza en la capa 7 del modelo OSI (capa de Aplicación) y continúa en todo el proceso hasta la capa 1 (capa Física). Por ejemplo, al enviar un correo ó transferir un archivo desde una computadora a otra, este mensaje ó archivo es sometido a una trasformación de un archivo discreto ó completo es convertido en muchos pedazos pequeños llamados paquetes. Empezando desde la capa de Aplicación del modelo OSI, el archivo continúa siendo dividido en pequeños pedazos hasta que el archivo discreto original sea, sea convertido en piezas suficientemente pequeñas, y más manejable pedacitos de información y entonces serán enviados a la capa Física.Cada capa agrega su propia información, llamada header, al paquete. Esta información permite que cada capa se comunique con las otras, y también permite que el computador recibiendo pueda procesar el mensaje.

***Creación de Paquetes- Remover el Header***

Ya hemos visto como el host que envía crea los paquetes. Cuando el host que recibe procesa un paquete, este efectúa lo contrario que el host que lo creo hizo y remueve casa header, empezando con el de la capa 1 (capa Física) y termina con al capa 7. Todo lo que queda al final del proceso es la data original sin ningún tipo de alteración, la cual puede ser procesada por el host que recibe.

## Protocolos de Aplicación, Transporte, y Red

Todos los protocolos discutidos en en este capítulo pertenecen a una de tres divisiones del modelo OSI. La primera categoría de protocolos abarca las capas del modelo OSI de Aplicación, Presentación, y Sesión. Estos se denominan como protocolos de la capa de Aplicación. Los protocolos de la capa de Transporte sólo aplican a la capa de Transporte; protocolos de redes abarcan las capas de Red, Enlace de Data y la física.

***Protocolos de la capa de Aplicación***

Los protocolos de la capa de Aplicación, también conocidos como protocolos de la capa alta, permiten que las aplicaciones interactúen entre si a través de la red. Los protocolos más comunes de la capa de Aplicación incluyen:

**Simple Mail Transfer Protocol (SMTP)**
: Protocolo de Transferencia de Correo Simple es un suite de protocolo TCP/IP usado para enviar mensajes de e-mail desde y a un host.

**Bootstrap Protocol (BootP)**
: Aunque aparentemente oscuro, el Protocolo de Bootstrap, el BootP es responsable de enviar paquetes TCP/IP dirigidos a la información de configuración del host.

**File Transfer Protocol (FTP)**
: Protocolo de Transferencia de Archivos también es parte del conjunto de herramientas TCP/IP, el FTP es usado para transferir archivos de un host a otro.

**Hypertext Transfer Protocol (HTTP)**
: El World Wide Web usa este conjunto o suite de protocolos TCP/IP para interconectar páginas Web.

**Simple Network Management Protocol (SNMP)**
: Suite de protocolos TCP/IP que permite que administradores de red diagnostiquen y administren redes sin importar la arquitectura de esta.

**Server Message Block (SMB) protocol**
: Protocolo usado en redes de Microsoft, este protocol permite que clientes trabajen con servidores, para accesar archivos y hacer peticiones de otros servicios. GNU/Linux puede interactúar con equipos usando el protocolo SMB utilizando el suite de Samba.

**Network File System (NFS)**
: Protocolo que permite compartir archivos e impresoras en redes UNiX.

***Protocolos de la Capa de Transporte***

La capa de Transporte provee transferencia de data confiable. Entre los protocolos usados en esta capa se incluyen:

**• Transmission Control Protocol (TCP)**
: Forma parte del conjunto de protocolo de TCP/IP, TCP provee transferencias confiable y administra las sesiones.

**• Sequenced Packet Exchange (SPX)**
: Parte del conjunto de protocolo IPX/SPX, el SPX es similar al TCP en que es quien administra las sesiones de comunicación.

**• NetBEUI**
: Permite que distintas aplicaciones en diferentes computadoras usando NetBIOS se comuniquen entre ellas; es un protocolo no enrutable. El uso más frecuente de NetBEUI es en redes de sistemas Microsoft.

***Protocolo de la Capa de Red***

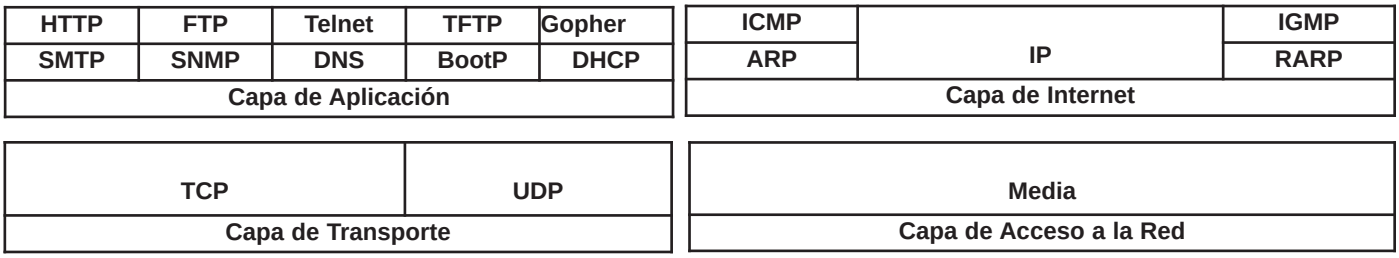
Los protocolos de Red proveén información de enrutamiento a los enrutadores y direcciones a los host.

Entre los protocolos de Red se incluyen:

- **Internet Protocol (IP)**
  Parte del conjunto TCP/IP, IP es el responsable de direccionamiento del host y de enrutar los paquetes en cualquier red ejecutando TCP/IP, incluyendo el Internet.
- **Internetwork Packet Exchange (IPX)**
  IPX provee servicios de direcionamiento para el suite de Novell IPX/SPX.
- **NetBEUI**
  Este permite que diferente aplicaciones en distintas computadoras usando NetBIOS se comuniquen una con otra; es un protocolo no enrutable.
- **Ethernet**
  Este protocolo LAN fué creado por Xerox, Digital Equipment Corporation, e Intel. Es la tecnología LAN más usada.

Estudiaremos más profundamente estos protocolos a través del libro.

## Protocolos de Internet




Cada capa de la arquitectura de Internet involucra protocolos, y cada protocolo tiene su RFC asociado. Esta sección, describiremos brevemente protocolos comúnmente usados en cada capa del Internet. Estos protocolos serán discutidos más detalladamente más adelante. Cada protocolo está listado con sus respectivos RFC(s).

### *La Capa de Acceso a la Red*

La capa de Acceso a la Red puede variar considerablemente, dependiente de que tipo de tecnología es responsable de colocar la data en el medio de la red y de captarla del medio. Algunos ejemplos:

| | |
|---|---|
| LAN | Ethernet, Token Ring, y Fiber Distributed Data Interfase (FDDI) |
| WAN | Frame Relay, serial lines, y Asynchronous Transfer Mode (ATM) |

### *La Capa de Internet*

Aquí listamos los diferentes protocolos de la capa de Internet y sus respectivos RFC:

**IP-RFC 791, STD 5**

IP es el método básico de transferir data usado en todo el Internet. Es el responsable de el direccionamiento de IP y ejecuta la función de enrutar, el cual selecciona una ruta para enviar la data a la dirección IP de destino.

La data se envía en formato de paquetes, también conocidos como datagramas. Un paquete es autocontenido, totalmente independiente de otros paquetes, que no requiere supervisión y carga suficiente información para enrutar desde el host de origen hasta el host de destino.

IP define como los enrutadores procesan los paquetes, cuando se deben generar mensajes de error, y bajo que condiciones los paquetes serán descartados.

**ICMP-RFC 792, STD 5**

ICMP es el protocolo de diagnóstico del TCP/IP. ICMP está especificado en los RFCs 844, 1256, y 1788. Este permite que los hosts y gateways (pasarelas) reporten mensajes de error a través de los mensajes ICMP. Si ocurre un problema en una red TCP/IP, un mensaje ICMP será generado.

**Internet Group Management Protocol (IGMP)-RFC 1112, STD 5**

El IGMP es usado para multicasting. En multicasting, una fuente encia un mensaje a un grupo de subscriptores (multicast

groups). Para que la transferencia de un multicast sea exitosa, los miembros deben identificarse y los grupos que les interesa deben hacerlo a un enrutador local. El IGMP le permite a los usuarios pertenecer y mantener membrecías en grupos multicast.

**ARP-RFC 826, STD 37**

ARP traduce las direcciones de Internet a direcciones físicas, como son las direcciones físicas de 48-bit de Ethernet.

Por ejemplo, asumimos dos hosts en una red, nodo1 y nodo2. Nodo1 sabe la dirección IP de node2. Pero si el node1 desea enviar un paquete al nodo2, este debe saber la dirección física, ó el hardware, de la dirección del nodo2. Para poder resolver la dirección IP a la dirección del hardware, el ARP envía un broadcast local y Así obtiene la dirección de hardware.

Una vez la dirección a sido resuelta, el ARP almacena la información para las peticiones futuras en un cache. El cache del ARP permanece por un período de tiempo que varía dependiendo del sistema operativo.

**RARP-RFC 903, STD 38**

Como su nombre lo implica RARP ejecuta la función en reverso del ARP. Este utiliza la dirección de hardware del nodo para requerir una dirección IP. RARP es por lo general usado durante la inicialización de una estación de trabajo sin discos para obtener su una dirección IP.

Por ejemplo, cuando un estación de trabajo se inicializa, el RARP leé la dirección de hardware única y envía un broadcast (difusión) sobre la red para pedir una dirección. Un servidor RARP responde a las peticiones y provee una dirección IP.

***La Capa de Transporte***

Aquí le presentamos los protocolos de la capa de Transporte y sus RFC que lo define a ellos:

**TCP-RFC 793, STD 7**

El TCP provee la administración de sesión entre los sistemas fuentes y destino. Este asegura que la data sea transportado en secuencia y que no se envíe data duplicada. El TCP es usado con aplicaciones que se comunican, estableciendo la sesión antes de transferir data, como son el FTP y el Telnet.

**UDP-RPC 768, STD 6**

El protocolo UDP provee una comunicación simple de paquete. Un paquete UDP es creado por cada operación de las aplicaciones, y una sesión no es necesaria. A diferencia del TCP, UDP no provee control de tráfico ni envía acuse de recibo. Este tampoco retransmite paquetes perdidos ni da garantía de confiabilidad. El UDP es un protocolo sin conexión que es usado por el TFTP (Trivial File Transfer Protocol) y SNMP.

***La Capa de Aplicación***

Los protocolos de la Capa de Aplicación y sus respectivos RFCs sin listados a continuación:

**HTTP-RFCs 1945 y 2616**

El HTTP es utilizado para transportar documentos HTML (páginas Web) a través del Internet. El HTTP requiere de un programa cliente (un navegador) y un servidor, ambos ejecutando TCP/IP. El HTTP establece la sesión del servidor Web y transmite páginas HTML a un cliente con un navegador. El protocolo HTTP 1.0 establece una nueva conexión por cada página requerida, lo cual creaba mucho tráfico innecesario de Internet. El HTTP 1.1 establece las conexiones persistentes, lo cual permite multiple descargas con una sola conexión. Ambos el cliente y el servidor deben soportar el HTTP 1.1 para poder beneficiarse de estas ventajas que ofrece esta nueva versión.

**FTP-RFC 959, STD 9**

FTP es un sistema para transferir archivos entre computadores ejecutándose en red TCP/IP. El FTP ofrece una manera eficiente y rápida de transferir archivos ya que este no tiene que consumir los recursos de codifica y decodificar la data, como es enviar archivos de correo electrónico con archivos adjuntos. El FTP permite que archivos sean copiados al servidor, mientras que HTTP sólo permite que los clientes descarguen desde el servidor.

**TFTP-RFC 1350, STD 33**

El TFTP se usa para iniciar los sistemas sin discos (diskless). Funciona con BootP. El TFTP utiliza UDP, mientras que el FTP usa TCP. Debido a que el TFTP es simple y pequeños, este puede ser embedido en Memoria de Sólo Lectura (ROM),

lo cual es ideal para las estaciones de trabajo sin discos que buscan configuraciones a través de la red para iniciarse.

**Telnet (remote terminal pro toco I)-RFCs 854 y 855, STD 8**

Telnet es un protocolo de emulación de terminal desarrollado por ARPANET. El telnet le permite a los usuarios desde un sitio ingresar a otro remotamente y ejecutar programas desde el sistema remoto. El Telnet es usado normalmente por el terminal cliente ó un emulador de terminal vía software en una PC.

**Network News Transfer Protocol (NNTP)-RFC 977**

NNTP permite que sitios en el Internet intercambien artículos de noticias de Usenet, los cuales son organizados en tópicos como Programación en C++ o en Perl. Para usar los newsgroups, deberá tener acceso a un servidor NNTP que usted este autorizado para leer y postear noticias.

**Gopher-RFC 1436**

Gopher es un programa basado por menu usado para encontrar recursos en el Internet. Es muy similar a lo que hoy día es la Web. Los usuarios siguen de vínculos a vínculo de sitio a sitio en búsqueda de su información. Fué una de las primeras herramientas desarrolladas para establecer el acceso a los usuarios de todo el Internet y no sólo a un sitio. Los servidores Gopher han sido reemplazados por Servidores Web.

**SMTP-RFC 821, STD 10**

SMTP es el protocolo estándar de Internet para transferir mensajes de correo electrónico desde un computador a otro. Este específica como dos sistemas de correos deben interactuar. El SMTP es usado a menudo con el protocolo POP3 (Post Office Protocol 3), Cual es el estándar servidor de correo de Internet que utilizan el protocolo de mensajería del SMTP. El POP3 almacena e-mail entrantes hasta que los usuarios se autentican y lo descargan. POP3 está definido en el RFC 1939, STD 53.

**SNMP-RFC1157, STD15**

El SNMP es usado para administrar redes TCP/IP. Es un esquema estandarizado de administración que los fabricantes pueden soportar. Todos los dispositivos que cumplen los estándares SNMP pueden ser administrados por un administrador de SNMP. El SNMP también ofrece un bajo nivel de requerimientos de recursos, portabilidad, y son ampliamente aceptados.

**Domain Name System (DNS)-RFCs 1034 y 1035, STD 13**

El DNS es un mecanismo usado en el Internet para traducir los nombres de computadores host a direcciones válidas de Internet (IP). Es un método universal centralizado de resolución de nombre. Por ejemplo, cuando un usuario hace una petición de un FQDN (Fully Qualified Domain Name) www.abiertos.org, los servidores DNS traducen el nombre a la dirección IP 206.123.67.148.

**BootP-RfCs 951, 1497 y 2132**

BootP es una alternativa al RARP. El BootP permite que estaciones de trabajo sin discos no sólo determinen sus direcciones IP, pero parámetros adicionales, como lo son el gateway por defecto y las direcciones particulares de los servidores DNS. A diferencia del RARP que sólo provee una dirección IP.

**Dynamic Host Configuration Protocol (DHCP)-RFC 2131**

El protocolo DHCP está basado en el BootP. Como el BootP, este está diseñado para asignar direcciones de Internet y los parámetros adicionales, como son los gateways por defecto y los servidores DNS, a nodos en una red TCP/IP. A diferencia del BootP, las direcciones y los parámetros del DHCP pueden ser cambiadas con el tiempo (por esto es el nombre de dinámico). Los Servidores DHCP pueden temporalmente alquilar direcciones y parámetros por un tiempo específico a un cliente, entonces reasignar la información a otro cliente cuando el alquiler expire.

## Multiplexado/Demultiplexing

Demultiplexing es el proceso que utiliza la computadora que recibe un paquete para poder procesarlo. Es una manera excelente de mostrar como es que el protocolo de Internet funciona dentro de la arquitectura de Internet. Aquí le mostramos una imagen para ilustrar este proceso.

## 99999999999999FOTO aquí

## Protocolos Especializado de Interfase Serial

La gran mayoría de usuarios accedan el Internet desde su casa usando un modem. El Punto de Presencia (The Point of Presence, POP) es donde el usuario marca vía su modem para accesar el Internet. Este POP es normalmente su Proveedor de Servicios de Internet (Internet Service Provider, ISP). También es el punto donde el proveedor de larga distancia se conecta a la compañía de servicio local. Si la compañía local no existe, entonces el the POP es la línea conectada al usuario.

Conexiones vía modem son normalmente hechas sobre una línea telefónica estándar y utilizan el protocolo de Punto a Punto (Point-to-Point Protocol, PPP) ó el protocolo de Internet de Línea Serial (Serial Line Internet Protocol, SLIP) para conectarse a su ISP. En la siguientes secciones se describen estos protocolos.

***Point-to-Point Protocol (PPP)-RFC 1661, STD 51***

El PPP es un método de encapsulación para enviar paquetes a través de un enlace serial. fué creado en el año 1991 por la IETF y soporta ambos tipo de enlaces asincrónicos y sincrónicos. Debido a esto es que se puede ejecutar sobre líneas telefónicas estándares, enlaces full-duplex como son las líneas de Redes Digitales de Servicios Integrados (Integrated Services Digital Networks, ISDN), y las líneas de alta velocidades como las T1 y T3.

El PPP utiliza el protocolo de Control de Enlace (Link Control Protocol, LCP) para establecer, configurar, y poner a prueba una conexión durante el proceso de login (ingreso al sistema). Este protocolo permite que ambas computadoras puedan negociar y provee un alto nivel de confiabilidad. El PPP también hace disponible la protección de contraseñas utilizando los protocolos Autenticación de Contraseñas (Password Authentication Protocol, PAP) y el CHAP (Challenge Handshake Authentication Protocol).

El PPP tiene una familia de protocolos específica Network Uyet protocols llamados Protocolos de Control de Redes (Network Control Protocols, NCP). Los NCP existen para IP, AppleTalk, y el DECnet. Un ejemplo es, el NCP de IP que permite a un host negociar los encabezados de compresión headers.

***Multilink Point-to-Point Protocol (MPPP)-RFC 1990***

Si un usuario se conecta a su ISP usando una línea estándar ISDN, el PPP por lo normal utiliza un canal B de transmisión de 64-Kbps. Para obtener velocidades más alta de transmisión, dos ó más canales B pueden ser compartidos usando MPPP. Un ejemplo es, dos canales B de tipo ISDN de 64-Kbps pueden ser combinados para transmisiones a velocidades de 128 Kbps.

***Serial Line Internet Protocol (SLIP)-RFC 1055, STD 47***

SLIP es una forma simple de encapsulación para enviar paquetes IP sobre líneas seriales. SLIP puede ser usado en puertos seriales RS-232 y es usado comúnmente para conectar usuarios domésticos (no empresariales) al Internet desde líneas telefónicas estándares. SLIP soporta enlaces asincrónicos. Scripts ya automatizados son por lo general utilizados para agilizar el proceso de ingreso al sistema ó login. SLIP ha sido reemplazado por PPP ya que este es más rápido y confiable.

## ENRUTAMIENTO DE TRAFICO

El proceso de identifica la ruta a una conexión a través del laberinto de las posibles rutas de conexiones en el Internet serán discutido en esta sección. Los paquetes deben ser enrutados de estación a estación hasta arribar a su destino. En esta sección discutiremos los protocolos que proveen el mecanismo para para la selección de rutas. Estos serán los tópicos a discutir:

- Protocolos Enrutables y No Enrutables/Routable and Nonroutahle Protocols
- TCP/IP
- IPX/SPX
- NetBEUI

• Redes de Multi-Protocolos/Multiprotocol Networks

**Protocolos Enrutable y los No rutables**

Algunos protocolos pueden pasar de LANs a WANs y más allá porque estas pueden direccionar y atravesar los enrutadores. Dentro de los protocolos enrutables están TCP/IP y IPX/SPX. Dentro de los protocolos no enrutable se incluyen el NetBEUI, el de Digital Equipment Corporation de nombre LAT (Local Area Transport), y el protocolo DLC (Data Link Control).

***TCP/IP***

El 1 de enero, 1983, las principales redes que conforman el Internet adoptaron el conjunto de herramientas (el Suite) TCP/IP como el protocolo oficial del Internet. Una de las razones del crecimiento explosivo del Internets y sus habilidades poderosas fué su adopción de esta suite, la cual originalmente fué desarrollada en la Universidad de Berkeley, en California.

Actualmente, el Internet soporta completamente la versión 4 del TCP/IP. La versión 6 del TCP/IP (mejor conocida como IPv6) ya se probó y ha adquirido soporte completo y esta siendo implementado en la actualidad. Discutiremos en detalle todo acerca del TCP/IP en todo este libro, algunos de los principios básicos lo cubriremos en esta sección.

***Todo una Colección de Protocolos***

El suite TCP/IP, es un conjunto de protocolos que incluye el Transmission Control Protocol (TCP), y el Protocolo de Internet (Internet Protocol, IP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), y muchos más. Existen un core de más de 20 protocolos en el suite completo, cubriendo todos los aspectos del uso y la administración de las redes. Cada una de estos protocolos tiene una función en específico. En esta sección sólo discutiremos el TCP y el IP.

***Transmission Control Protocol (TCP)***

El TCP es el Protocolo de Control de Transmisión que garantiza una comunicación confiable y utiliza puertos para transportar paquetes. Además también fragmenta y reemsambla los mensajes, usando una secuencia de funciones para asegurarse que los paquetes sean reemsamblados en el orden correcto.

***Internet Protocol (IP)***

El IP es un protocolo sin conexión responsable de proveer direccionamiento de cada computadora y ejecución de enrutamiento. La Versión 4 del TCP/IP usa direcciones de 32-bit. El esquema de dirección cae en 5 clases, solo tres de cada uno esta disponible por el direccionamiento de redes estándares. El plan original fué asignar direcciones de clase A a grandes redes, Clase B a redes de tamaño, y Clase C para pequeñas redes. Las direcciones de Clase D son usadas para multicasting, y las direcciones de clase E son experimentales. (Más adelante se discutirá más sobre las clases de direcciones.)

Direcciones IP de 32-bit están divididas en dos partes: la porción de red y la porción de host. La mascara de subred ayuda a determinar cuales bits son de subred y de hosts.

***Un estándar Abierto***

El TCP/IP no esta atado a ningún vendedor y, por lo tanto, permite redes heterogéneas se comuniquen eficientemente. Las implementaciones están disponibles para un gran rango del sistema, al extenderse desde las redes de computadoras, Asistentes Digitales Personales (PDAs), laptops, y de computadores personales a mainframes. Este usa el modelo de arquitectura de Internet que divide este protocolo en cuatro capas. Cada capa es responsable de una tarea de comunicación y coinciden con las capas de OSI.

• Capa de Aplicación

- Capa de Transporte
- Capa de Internet
- Capa de Acceso a la Red

***El TCP/IP en Acción***

Una de las responsabilidades del IP es enrutar la información entre dos host entrelazados por Internet. Esta tiene el conocimientos de cual host enviarle los datos acortando la distancia de la manera más eficiente.

Foto de dos computadora conectadas mostrando las capas del tcp/ip

Si los dos puntos se encuentran físicamente en la misma red, entonces la comunicación es mucho más director; y no hay necesidad de la funcionalidad de enrutar.

***TCP/IP Sobre GNU/Linux y UNiX***

En sus inicio el desarrollo del TCP/IP fué muy cercano al de BSD UNiX, y la primera implementación apareció en la versión 4.2 BSD UNiX al principió de los 80s. La razón de esto es que ambos códigos fuente, el de UNiX y las especificaciones del protocolo TCP/IP, estaban disponible en toda la comunidad académica de Norteamérica. Los investigadores combinaron los protocolos TCP/IP con el sistema operativo UNiX para crear lo que se denominó un sistema operativo listo para el Internet.

Desde ahí al día de hoy la popularidad del TCP/IP ha crecido a proporciones gigantescas, y se ha convertido en el sistema de redes por defecto de UNiX. Combinandolo con la capacidad y el soporte que UNiX ofrece de Multiprocesadores Simétricos (Symmetric Multiprocessing, SMP) y su característica de multitarea (multitasking) hacen una combinación poderosa. De hecho, hoy en día todas las versiónes de UNiX vienen con TCP/IP. Los sistemas UNiX son en la actualidad la plataforma más usada como servidores de páginas Web.

UNiX es reconocido por su confiabilidad y es escogido las mayoría de las veces para los sistemas que ejecutan aplicaciones de misión crítica. En el pasado, entre las desventajas de UNiX se destacaban dos una era su alto costo y la otra era su falta de lo denominado userfriendly ó fácil uso. Pero recientemente, esto ha ido cambiando, clones de licenciamiento libre y/o abierto de UNiX, de los cuales GNU/Linux es sin lugar a duda es la versión más popular. Estas versiónes free/opensource y con calidad de nivel paralelo a los UNiX tradicional, ayudan a combatir la situación de costo. Se han ido elaborando más y más aplicaciones gráficas de toda indole, que ayudan en la situación del fácil uso y rápido aprendizaje. Hasta en el á de administración ya existen varios front-ends ó GUIs gráficos que hacen de la administración básica de sistemas GNU/Linux tareas mucho más fácil que tradicionalmente.

GNU/Linux con su kernel monolítico Linux y con la pila TCP/IP es uno de los sistemas operativos más rápidos disponibles en la actualidad. Además el kernel Linux también tiene excelente soporte de procesadores SMP para equipos con multiprocesadores. Por estas razones es que servidores GNU/Linux se ha ido ganando tanto espacio en el mercado de servidores Web, además de otros tipos de aplicaciones intensas de redes.

**IPX/SPX**

La compañía Novell, Inc., desarrolló el una vez dominante protocolo de LAN y WAN. Similar al TCP/IP, el IPX/SPX es un suite de protocolos y no un sólo protocolo. Sistemas operativos de Microsoft también ofrecen soporte de IPX/SPX, aunque la compañía lo ha renombrado NetWare Link (NWLink) por razones legales.

El IPX es un protocolo de la capa de Red sin conexión el cual es responsables del reenvío de los paquetes (forwarding packets). Los sistemas que se comunican entre si a través de una red IPX/SPX deben saber el número de red del host remoto Así como la dirección del sistema en la red. Finalmente ellos deben saber el número de socket del programa con el cual ellos se comunican. El IPX permite cada una de esta actividades.

999999999999999999999999

**SPX**

El SPX es un protocolo de la capa de Transporte que usa los Servicios proveídos por el IPX. Como este reside en la capa de Transporte, este se asegura el transporte de la data seguro y confiable y además administra las sesiones.

***Ventajas y Desventajas***

El IPX/SPX no es un protocolo neutral de fabricante. Fué desarrollado por Novell, usado mayormente en redes de Novell NetWare. El TCP/IP ha desplazado totalmente el IPX/SPX como el protocolo estándar de las redes empresariales debido a su naturaleza abierta. Pero, el IPX/SPX está aún en uso y todo profesional de las redes e ingeniero de sistemas deben aún conocerlo, recuerde que siempre a presentado un mejor perfomance que el TCP/IP.

Aunque el IPX/SPX no e soportado por el Internet, existen miles de redes WAN ejecutándose sobre redes IPX/SPX privadas ó Redes Virtuales Privadas (Virtual Private Networks, VPN) para comunicarse sobre largas distancias.

Ya Novell ha adoptado el TCP/IP como su protocolo por defecto en las versiónes de Novell NetWare 5, aunque aún soporta el protocolo IPX/SPX.

**NetBEUI**

El protocolo NetBEUI es un acronismo de Network Basic Input/Output System (NetBIOS) Extended User Interfase. Primero fué desarrollada por IBM, más luego Microsoft la implementó como solución para sus redes peer-to-peer. Es un protocolo no enrutable, lo cual limita su uso en muchas redes.

***NetBIOS***

El NetBIOS fué originalmente diseñado para ser usado con NetBEUI (por esto lo del nombre NetBIOS Interfase de Usuario Extendido). Como el NetBEUI esta disminuyendo su popularidad, NetBIOS es usado principalmente como una interfase de de programación de aplicaciones. Este reside en la capa de Sesión (Layer 5) del modelo OSI. El NetBIOS puede operar sobre el NetBEUI Así como también como con protocolos enrutables como es el TCP/IP y IPX/SPX. Los equipos ejecutando Microsoft Windows usan nombres NetBIOS para identificarse entre si en la redes.

**Redes Multiprotocolo**

Las redes comúnmente usan protocolos enrutables como son el TCP/IP y el IPX/SPX, aunque esta combinación puede causar problemas con sobrecarga de sistemas, en sitios webs muy visitados. Tales combinaciones de sistemas proveén redundancias y pueden así acelerar su velocidad.

Hay veces que es beneficioso combinar protocolos enrutable y no enrutable , aunque se encuentre en una red enrutable. Un protocolo no enrutable como lo es NetBEUI puede resultar muy útil en situaciones LAN y WAN ya que este puede transportar tráfico a una computadora local sin la problemática asociadas del TCP/IP. Si un usuario envía un mensaje a un empleado en la misma LAN, entonces el NetBEUI controlará

toda esta transacción. pero, si alguien envía un mensaje a alguien en otra LAN (actividad que obviamente involucra un enrutador), El sistema automáticamente usará un protocolo como lo es el TCP/IP.

## RESUMEN

En este capítulo discutimos los siguientes tópicos de redes:

- El concepto de redes y su rol en la intercomunicación de data en las empresas hoy día
- Como se relacionan el Internetwork (trabajo a través del Internet) al concepto de redes corporativas empresariales
- Servidores, estaciones de trabajos, y hosts
- características Básicas de las topologías de Redes
- Sistemas Operativos de Redes (NOS)
- Puntos básicos de las Redes de Area Local (LAN) y las Redes de Area Amplia (WAN)

## PREGUNTAS POST-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿De cuantas capas consiste el modelo OSI? ¿Cual es la capa que más se relaciona con hardware?
2. ¿Que son los RFCs?
3. Liste tres protocolos comunes del Internet y los puertos que generalmente usan.

# CAPITULO 2

*REDES DEL SISTEMA OPERATIVO GNU/LINUX*

# FUNDAMENTOS DE IP


| TÓPICOS PRINCIPALES | No. |
|---|---|
| Objectivos | 68 |
| Preguntas Pre-Examen | 68 |
| Introducción | 69 |
| Componentes Comunes de Red | 70 |
| Medios de Transmisión | 81 |
| Tipos de Transmisión | 89 |
| Capa de Enlace de Data | 92 |
| Estándares de Comunicación | 110 |
| Direccionamiento de Internet | 132 |
| Interfases de Red | 155 |
| El Futuro del IPv6 | 176 |
| Resumen | 190 |
| Pregunta Post-Examen | 191 |


## OBJECTIVOS

Al finalizar este capítulo, usted estará preparado para efectuar las siguientes tareas:

- Usar el TCP/IP para efectúar funciones básicas.
- Definir la importancia de los archivos de inicio del sistema y los pasos necesarios para accesarlos.
- Entender el rol de los scripts y los archivos de configuración en /etc/sysconfig/network-scripts.
- Configure y use el Protocolo PPP (Point-to-Point Protocol).
- Describe el sistema de direcciones de IP así como el nuevo sistema que ha sido desarrollado para poder enfrentar el crecimiento exponencial del Internet.
- Describa la herramienta de red WvDial y su rol en la configuración de una red.

## PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuál es la función del daemon inetd?
2. ¿Cuáles son los tres tipos de topologías lógicas de redes?
3. ¿Cuáles dos protocolos pueden ser usados para conectarse al Internet vía modem? ¿Cuál es mejor, y por qué?
4. ¿Cuál es el rango de las tres principales clases de direcciones?
5. ¿Por qué existe la necesidad del IPv6?

## INTRODUCCIÓN

El Protocolo de Control de Transmisión (Transmission Control Protocol/Internet Protocol, TCP/IP) es el protocolo principal de redes usado por todos los sistemas GNU/Linux. Es utilizado por todas las computadoras que desean comunicarse en el Internet con otras computadoras. También puede ser usada para intercomunicar computadoras en redes privadas e intranets.

Para que los host se puedan comunicar con otros hosts remotamente sobre el Internet, deberá saber la dirección de Internet. Cada host, ó nodo, tiene su propia dirección Internet de 32-bit, ó dirección IP, que la identifica como equipo único y diferente de cualquier host en el Internet.

Este capítulo se inicia describiendo las capas con que cuenta el TCP/IP: Las capas Física y la de Enlace de Data del modelo OSI. Identificaremos los componentes necesarios que permiten que LANs y WANs funcionen. Luego describiremos las tecnologías de redes más comunes como son el Ethernet, Token Ring, PPP, y Serial Line Internet Protocol (SLIP). También cubriremos la versión actual del protocolo de Internet (IP) para aplicar direcciones de Internet y las mayoría de redes TCP/IP de hoy, el IPv4, y el del futuro el IPv6. La estructura de las direcciones de Internet, el formato binario contra el formato decimal, clases de direcciones, reglas de direccionar, direcciones reservadas, subredes (subnetworks), mascaras de subredes, y rango de direcciones.

## COMPONENTES DE REDES COMÚN

Redes de computadoras por lo normal requieren un conjunto de equipo para poder funcionar apropiadamente. Típicamente las LANs y WANs incluyen Tarjetas de Interfase de Red (Network Interfase Cards, NICs), repetidores, hubs, bridges, enrutadores (routers), brouters, y switches.

Cubriremos los siguientes componentes en esta sección:

- **Network Interfase Cards (NICs)**
- **Repeaters**
- **Hubs**
- **Bridges**
- **Routers**
- **Brouters**
- **Switches**
- **Gateways**
- **Channel Service Unit/Data Service Unit (CSU/DSU)**
- **Modems**
- **Patch Panels**
- **Internet in a Box**

### Network Interfase Cards (NICs)

Cada nodo en la red debe tener una tarjeta llamadas NIC, a menudo referenciadas como tarjeta de red. El NIC es el interfase entre el computador y la red. Hoy día la mayoría de tarjetas madres ó motherboards, la traen integrada, pero es más común verla como tarjetas de expansión PCI. Estas se comunican con otras computadoras a través del NIC y su manejador. Redes se conectan a través de un cable conectado a la NIC y de ahí a la red. Los NICs varían en su tipo de tecnología como son redes Ethernet y Token Ring, las cuales estudiaremos en el próximo capítulo. Las NICs operan en la capa de Enlace de Data del Modelo de OSI.

La mayoría de los NICs contienen un transductor, un dispositivo de red que transmite y recibe señales análoga ó digital. La terminología es de transmitor y recibidor. En las LANs, los transductores colocan data

en el cable de la red y detecta y recibe la data que se traslada a través del cable. Algunos tipos de redes requieren un transmitor externo.

**Repeaters**

Un repetidor es un dispositivo de bajo nivel que amplifica la señal electrónica que viaja en el segmento de un cable. Nos asegura que la señal electrónica no degrade. así que un repetidor puede conectar dos computadoras que se encuentra a una distancia mayor que la definida por los Estándares, así como Ethernet.

Los repetidores operan en la capa Física (Layer 1) del modelo OSI. Ellos transmiten código binario, los cuales consisten en unos y ceros. Un segmento de cable que agota su longitud máxima empieza a degradar su señal y eventualmente se colapsa. Un repetidor puede reforzar la señal a través de su retransmisión.

**Hubs**

Un hub conecta las computadoras en una configuración de red tipo estrella (star) para que ellas puedan intercambiar información. Tienen varios puertos disponibles, cada uno conectado a un nodo único. a través de esta interconexión de los nodos, un hub sirve como el punto de concentración de una red. La mayoría de los hubs son conocidos como hubs activos porque ellos regeneran señales electrónicas al igual que los repetidores.

Los hubs operan en la capa Física (Layer 1) del modelo OSI. Ellos se pueden conectar a otro hubs, y conectados en cadena para poder tener más puertos disponibles para acomodar redes más grandes. Usted puede conectar hubs a los switches ó a los routers para incrementar el número de nodos.

**Bridges**

Los bridges son dispositivos que filtran las tramas (frames) para determinar si un frame en específico pertenecen a un segmento local ó a otro segmento de LAN. Como los bridges operan en la Capa 2 del modelo OSI, ellos usan direcciones de hardware para determinar cual segmento recibirá la trama.

Los bridges pueden reducir el tráfico de la red dividiendo una red en dos segmentos. Ellos también pueden conectar segmentos de redes con el mismo ó diferente protocolos de enlace de data, habilitandolo para comunicarse. Por ejemplo, un bridge puede conectar una red de Ethernet con una de Token Ring ó conectar dos redes Token Ring.

Los bridges reconocen hardware, ó direcciones de Control de Acceso al Medio (Media Access Control, MAC), entre las redes. Supongamos que un computador envía información a otra en una red tipo Ethernet. El bridge determina si la computadora de destino reside en el mismo segmento de red a través de la verificación de la dirección MAC. Si el computador de destino reside fuera del segmento, el bridge pasa el mensaje a otro segmento. a través de este proceso de filtrado, el tráfico de la red es reducido y se libera más ancho de banda y se hace disponible.

Los bridges son independiente de todos los protocolos de las capas superiores. Esta independencia permite a los bridges poder reenviar tramas que provienen desde muchas diferentes capas de protocolos.

**Enrutadores/Routers**

Los routers son conceptualmente similar a los bridges excepto que estos obedecen la capa 3 del modelo (Layer 3) OSI. En ves de usar la dirección MAC, los routers usan protocolos de redes, como es el IP y el IPX. Ellos reenvían, o enrutan, data desde una red a otra en vez de un segmento a otro.

Los routers direccionan los paquetes de data entre las redes. Ellos identifican la dirección de red de la

computadora de destino, y entonces determinan la ruta más eficiente para transmitirle la data.

Suponga que una máquina en la red 2 envía data a otra máquina en la red 2. En este caso, el router no transmitirá la data a la red 1. Este proceso de filtrado conserva en ancho de banda de la red. Se puede medir la capacidad de un router a través de su rango de paquetes por segundo (packets per second,PPS).

Los routers dependen del protocolo—ellos dependen del sistema de direccionamiento definido por el protocolo usado (IPX, IP, entre otros). Diferentes tipos de routers trabajan con diferentes protocolos. Por ejemplo, los IP routers operan con la estructura direccional del inherente IP de 32-bits. Para usar el protocolo de direccionamiento IPX, es necesario un router que soporte tanto IP como IPX. El enrutamiento se explicara com más detalle en este capítulo.

**Brouters**

Los brouters operan en las capas 2 y 3 del modelo OSI. Ellos incorporan la funcionalidad de los bridges y routers a la vez. Los brouters son protocolos independientes y pueden enrutar el tráfico de varias redes. Por ejemplo, ellos pueden enlazar paquetes DECnet y enrutar un paquete TCP/IP.

**Switches**

En una red, un interruptor (switch) direcciona el flujo de información de un nodo a otro. Los Switches operan más rápido que los dispositivos de red tradicionales, tales como los hubs, bridges, y routers, y están progresivamente reemplazando estos dispositivos. Un switch puede asignar a cada par de emisor/receptor el ancho de banda total de las líneas en lugar de compartir el ancho de banda con todos los demás nodos de la red.

Los switches ofrecen los siguientes beneficios para las redes Ethernet y Token Ring:

- **Instalación sencilla**
  Para la mayoría de los bridges y hubs, al instalar un switch se requiere que usted desconecte desde los dispositivos existentes y luego reconecta todos los dispositivos en los puertos del switch.
- **Alta Velocidades**
  Los witches tienen backplanes de alta velocidad (que son las conexiones dentro del switch mismo) que permiten banda ancha completa entre cualquier dos usuarios o cualquier otro dos segmentos. Esta característica elimina el switch como un potencial cuello de botella en la red.
- **Más Ancho de Banda en los Servidores**
  Los servidores se pueden conectar directamente a los switches. Esta capacidad permite que los usuarios de la red usen todo el ancho de banda al accesar los recursos del servidor.

Los switches pueden operar en varias capas del modelo OSI. Un switch de Capa 1, llamado un hub de switching, sirve para reemplazar los tradicionales y mucho más lento hub. Un switch de Capa 2, también llamado un LAN switch, reenvía tráfico basado en la dirección MAC. Un switch de Capa 3 reenvía tráfico basado en la información de la capa 3y se llama un routing switch y soporta protocolos de red, como son IP y el IPX. Los switches de Capa 4 toman desiciones de reenvío basadas en la información de capa 4, como puede ser el puerto específico TCP/UDP que una aplicación usa, así como la información de las capas 2 y la 3.

***Las opciones para manejar un incremento del tráfico de red de una LAN***

A continuación tres opciones que pueden manejar el incremento de tráfico en la LAN:

- **Usar un bridge.**
  Este método tradicional reduce el número de usuarios en una red a través de separarla en dos segmento de red.

- **Usar un LAN switch (un switch de Capa 2)**
  Los LAN switches están disponibles para Ethernet, Fast Ethernet, Token Ring, y FDDI (aprenderemos más de estos Estándares de red más adelante en el próximo capítulo).

**• Incrementar el ancho de banda de la red**

Una forma de incrementar ancho de banda es mudarse hacia Estándares de más alta velocidad como es el estándar FDDI. Para lograr estos cambios hay que cambiar adaptadores, re-alambrar todas las conexiones, y es muy posible tener que cambiar software del sistema. Actualizar a FDDI es costoso y en algunas situaciones hasta caro. Actualizar de Ethernet a Fast Ethernet es mucho más económico. Aprenderemos de FDDI y Ethernet en este capítulo.

## Gateways

Los Gateways, que también son conocidos como convertidores de protocolos, pueden operar en cualquier capa del modelo OSI. El trabajo de un gateway es mucho más complejo que ese de un router o switch. Normalmente, un gateway deberá convertir desde una pila de protocolo a otra. Por ejemplo, un gateway puede interconectar un nodo de red AppleTalk a un nodo DECnet o a un nodo de red TCP/IP o a un nodo de red IPX/SPX. Discutiremos otro tipo de gateway, el gateway por defecto, más adelante.

## Channel Service Unit/Data Service Unit (CSU/DSU)

Un CSU/DSU (Unidad de Canal de Servicio/Unidad de Servicio de Data) es donde terminan las conexiones físicas. Este dispositivo es requerido cuando usamos circuitos dedicados, como son las líneas T1. El flujo digital de data es traducido por el CSU/DSU a señales bipolares, las cuales son adecuadas para la transmisión por línea.

Por ejemplo, una línea dedicada Tl entra a un edificio a través de un conector RJ-45 (Registered Jack-45), el cual es parecido a un conector de teléfono pero más grande (los jacks de teléfonos usan conectores RJ-11). El CSU/DSU entonces transmite la señal a la red.

El CSU/DSU también lleva acabo cierto reporte de errores y funciones de loopback. Los CSU/DSUs opera en la Capa Física (Capa 1) del modelo OSI.

## Modems

Un modem es un dispositivo que permite que un computador se comunique con otras computadoras sobre una línea telefónica a través de la traducción de data digital a señales audio/análogas (en la computadora que envía) y de nuevo a data digital en la computadora que la recibe. Este tipo de modem es denominado tradicionalmente como modem análogo o simplemente modem.

El término modem es usado en una manera muy general y no siempre se refiere a modems que traducen análogo-a-digital y vice versa, por ejemplo otros son modems del tipo cable, DSL y los ISDN que son usados en todas las redes digital; la traducción a análoga no es requerida ni necesaria. La terminología modem a sido usada para describir cualquier dispositivo que adapta una computadora a una línea telefónica o a redes de Telecable, aún sea digital o análoga.

Un modem puede ser compartido en una red con todos los usuarios con acceso a ella. La computadora que se encuentra conectada físicamente al modem debe estar configurada apropiadamente. Al configurar un modem para acceso acceso por una WAN (o cualquier tipo de actividad o propósito de red), deberá definir varios ítemes para asegurar un funcionamiento exitosos. Estos ítemes incluyen:

- IRQ del Puerto Serial
- Dirección de E/S (I/O Address)
- Velocidad máxima del Puerto

### IRQ del Puerto Serial

Una línea de Interrupt Request (IRQ) es usada por componentes del tipo modems, tarjetas de redes, y los teclados para requerir atención del procesador del sistema. Los modems seriales pueden usar una de las líneas de IRQ 3 o 4, ambas son usadas para puertos seriales.

***I/O Address***

Las direcciones de Entra/Salida (input/output, I/O) transfieren información entre el CPU y un dispositivo en específico. La configuración de los puertos base de I/O de un modem son 3FO al 3FF para el COM1 y la 2FO a la *2FF* para el COM2.

***Velocidad máxima del Puerto***

Los modems están disponibles con diferente capacidades para enviar y recibir datos. Al configurar su modem, usted debe establecer la velocidad máxima del modem para asegurar que funcione apropiadamente. Esta configuración provee la conexión más confiable para el modem y de optima velocidad de transmisión para su sesión.

***Patch Panels***

Los Paneles de Patch agrupan los sockets (normalmente consisten en pin de conexiones y puertos) montados en un rack. Es un punto central donde los diferentes cables desde los diferentes puntos de la red (digamos los departamentos) pueden ser interconectado (así formando una LAN). Puede ser usada para conectar una red completa al Internet o a otra WAN.

Los patch panels son generalmente colocados en un punto central, en sitios como en un armario o en cuarto de máquinas de la compañía. Ellos pueden tener varios puertos y los pines de conexión dependiendo del tamaño de la compañía.

En un lado el patch panel consiste de filas de pines de conexión. Los cables que se conectan a los pines de conexión se introducen con una herramienta de empuje (punch tool) para hacer la conexión. Estas conexiones por lo general se originan desde jacks en las paredes de la oficina. Por ejemplo, el departamento nómina con 35 estaciones de trabajo puede conectar cada estación de trabajo a un hub. El hub entonces es conectado a un jack en la pared, el cual es entonces conectado al patch panel.

El otro lado del patch panel contiene una fila de puertos de conectores hembra, los cuales son usados para conectarse a otros dispositivos de red, somo son los routers y los switches. Por ejemplo, el patch panel puede ser conectado a un router, el cual entonces es conectado al Internet.

Los Patch cords son usado en los puertos para conectar crusado computadoras en red que están conectadas al patch panel. Los patch cords conectan dispositivos de red a un jack de pared (wall jack).

**Listo para el Internet**

GNU/Linux ofrece distribuciones que ofrecen la oportunidad, a pequeñas empresa y usuarios, de soluciones toda incluido de Internet. Distribuciones especializadas que ofrecen todo el software necesario para el acceso a Internet. También existen productos llamados soluciones TurnKey que incluyen además del software también el hardware. Por ejemplo estas soluciones pueden que incluyan servidores de correo, páginas web entre otras cosas. Este tipo de soluciones son de las más demandadas en comunidades en paises donde se empieza a desarrollar el uso de GNU/Linux y aún existen pocos expertos.

## MEDIO DE TRANSMISION

Para transmitir data, debe existir un medio físico, generalmente un cable o por el método común de hoy día wireless. En esta sección explicamos los tipos de cables más comunes:

• Cable Twisted-Pair (Trenzados)

- Cable Coaxial
- Cable de Fibra-Óptica
- Medio Inalámbrico/Wireless Media

**Cable Twisted-Pair**

El tipo de cable Twisted-Pair es quizás el más usado sistema de cable en las redes de Ethernet. Dos alambres de cobre trenzados entre si para formar un par de cables entre-tejidos. Dependiendo de la categoría, varios hilos de alambre aislados pueden residir en un sólo cable. Los diferentes tipos de categorías de cable serán discutidas más adelante en este capítulo.

Un segmento de cable trenzado no puede sobrepasar los 100 metros. Los cables Trenzados son usados en muchos de los Estándares de redes. Por ejemplo, Redes Ethernet 10BaseT usan cable Trenzados; el nombre 10BaseT significa una red transmitiendo a 10 Mbps usando transmisión en banda base y cable par-trenzado. Los cable trenzado están disponibles en dos formas básicas:

**Trenzados Blindados/Shielded Twisted Pair (STP)**
Este es un trenzado de alambre de cobre protegido de interferencia electromagnética externa por un blindado de metal que cubre todo el alambre; es más dificil de instalar y mantener que el UTP.

**Trenzados sin Blindaje/Unshielded Twisted Pair (UTP)**
El tipo de cableado UTP es el más común de los cables trenzados; es más económico que el STP pero menos seguro y subsetible a las interferencias electromagneticas. En este libro sólo nos concentramos en el UTP.

Alambre STP y UTP están disponible en dos variedades:

- **Stranded**
  Este es el tipo más común; es flexible y fácil de manejar en las esquinas y los objetos.
- **Solid/Sólido**
  Este tipo puede usarse en distancias más grande sin desarrollar perdidas como el del tipo stranded, pero es menos flexible; se romperá si se dobla multiple veces.

Hay cinco Estándares de cables trenzados especificados por la Telecommunications Industry Association/Electronic Industries Alliance (TIA/EIA) 568 Commercial Building Wiring standard. Dos niveles adicionales, las Categorías 6 y la 7, también son comerciales pero no estandarizadas aún (la 6 ya si lo es) pero son usadas comercialmete. En la siguiente tabla se definen estas categorías.

| Categoría | Descripción |
|---|---|
| **1** | Usado para voz, no para data (UTP only) |
| **2** | Contiene cuatro trenzados y una transmisión de data hasta 4 Mbps; usado para redes Token Ring (Sólo UTP) |
| **3** | Contiene cuatro trenzados y una transmisión de data hasta 10 Mbps; usado para Ethernet |
| **4** | Contiene cuatro trenzados y una transmisión de data hasta 16 Mbps; usado para redes Token Ring |
| **5** | Contiene cuatro trenzados y una transmisión de data hasta100 Mbps; usado para Ethernet y Fast Ethernet y permite que Ethernet sea fácilmente actualizada a Fast Ethernet |
| **6** | Contiene cuatro trenzados y una transmisión de data hasta 155 Mhps; usado para Fast Ethernet |
| **7** | Contiene cuatro trenzados y una transmisión de data hasta 1,000 Mbps; usado para Ethernet de Gigabit |

***Conectores RJ-45***

Los conectores RJ-45 son comúnmente usados ciertos tipos de redes Ethernet y Token Ring, las cuales discutiremos más adelante en este mismo capítulo. Los conectores pueden sostener hasta ocho alambres y es usada con alambres trenzados. Para colocar un conector RJ-45 a un cable, debe hacerse usando una herramienta especial llamada crimper tool. Para colocar el conector, primero debe poner el cable con los alambres en posición correcta (la posición del alambre depende en cual de los estándar esta usando, lo discutiremos más adelante en este mismo capítulo). Luego coloque el conector en la herramienta crimper incluyendo el cable y simplemente oprima la herramienta firmemente.

El crimper empuja dos grapas desde el conector RJ-45 dentro del cable. Una grapa se inserta dentro del exterior del cable (el jacket) y así establece una firme conexión, entre conector y el cable. La otra grapa empuja ocho pines trapazando el cobertor del cable y lo introduce dentro del conector cada respectivo alambre cayendo en una canaleta.

Un conector RJ-45 es un poco más grande que un conector estándar de teléfono RJ-11, el cual solo puede recibir cuatro alambres. (Existe un RJ-11 de seis alambres también.) Un conector de 50 pines RJ-21, es un conector normal RJ-45 de un lado que se expande a doce (12) conectores RJ-45 del otro lado.

**Coaxial Cable**

El Cable Coaxial, mejor conocido como coax, es un cable de alta capacidad usado para redes de video y comunicación. Este proveé anchos de banda más altos que eso proveídos por cables par-trenzados. Los cables Coaxiales contienen en el centro un cable de señal, el cual puede ser hebras tejidas (stranded) o solid, envuelto el una malla de protección metálica que a la vez le sirve de tierra. Estas protecciones son o del tipo malla tejidas o solidas y son envuelta por una camisa plástica. Si el cable viaja por un plenum, este entonces es cubierto por un material a prueba de fuego, como es el Teflon.

Existen varios tipos de de cables coaxiales para diferentes propósitos. Por ejemplo, los cables coaxiales fueron diseñados para baseband, broadband, y redes de televisión.

***Cable Coaxial Grueso/Thick Coaxial Cable (Thicknet)***

El cable Thick Coaxial, o Thicknet, es considerado el estándar de Ethernet. Es muy a menudo referido como el cable amarillo, aunque el cable esta disponible en una gamma de colores. El nombre Thicknet se refiere a a uno de los varios métodos de conectar computadoras a un ambiente de red. Es usado para redes Ethernet 10Base 5 (Redes Ethernet funcionando transmisiones en una banda de 10 Mbps y cable grueso coaxial de 1/2 pulgadas de diámetro).

El cable Thicknet trabaja bien en ambientes donde interferencias de magnetismo pueden afectar el cable; su uso es muy común en hospitales donde se usan equipos de Computadoras para asistir diagnóstico como son los CADScan y los equipos de resonancias magnéticas o scanners para producir imágenes (MRI y CT). El cable no es muy flexible, por esto dobla fácil en esquinas. Cables Thicknet pueden ser tendidos en segmentos de un máximo de 500 metros.

***Cable Coaxial Fino/Thin Coaxial Cable (Thinnet)***

El cable Thin coaxial, o Thinnet, es un estándar Ethernet para redes pequeñas. El Thinnet es altamente flexible y trabaja a perfección en á reducidas, como son oficinas divididas por cubículos. Un segmento Thinnet puede ser tendido hasta un máximo de 185 metros, pero con dispositivos de de red (como es un repetidor) esta distancia puede ser extendida. Thinnet es usado para redes Ethernet 10Base2 (Redes Ethernet ejecutándose transmisiones sobre una banda de 10 Mbps y un cable thin coaxial de 1/4 pulgada en diámetro).

***Conector BNC***

El conector BNC (British Naval Connector o Bayonet Neill-Concelman, BNC) es comúnmente usado para conectar cables coaxiales a NICs, hubs, y otros dispositivos de red. El BNC es engrapado al cable usando una técnica de nombre bayonet mount. La técnica bayonet mount conecta dos alambres (señal y tierra) en el cable coaxial al conector. El conector es luego insertado dentro de otro conector y luego enroscado, lo cual causa que el mecanismo bayonet pinche varios pines en la ranura del BNC.

***Cable de Fibras-Ópticas/Fiber-Optic Cable***

Cable de fibra-Ópticas pueden acomodar transmisión de data a velocidades muchas más alta que las de cables coaxiales o par-trenzados. Líneas de fibra-Ópticas pueden transmitir data a velocidades de en los rangos de gigabits por segundos. Ya que ellas envían data como pulsos de luz sobre hebras de vidrio, la transmisión puede viajar millas sin degradación de la señal. Como es un rayo de luz no se transmite ninguna señal eléctrica sobre la fibra-Ópticas, así que las líneas son libre de interferencias electromagnéticas y son extremadamente difíciles de interferir.

Cables de fibras-Ópticas consisten en dos hilos de vidrio pequeños; uno que recibe y el otro que envía. Estas hebras de hilo son llamadas el núcleo (core), y aveces son echas de plásticos. Cada core es rodeado por vidrio cladding. Cada core y elemento cladding está envuelto con una fibra Kevlar de plástico reforzado.

Transmisores láser envían los pulsos de luz modulada, y recibidores ópticos los reciben. Existen dos principales tipos de cables de fibra Ópticas:

- **Single mode**
  Este usa una longitud de onda específica de luz (wavelength). El diámetro del core de este cable es de 8 a 10 micrónes. Fibras de Mono Modo es muy comúnmente usado en troncales telefónicas en el interior de ciudades y aplicaciones de video.
- **Multimode**
  Este usa un número mayor de frecuencias (o modos). El core del cable es más grande que el del mono modo. Fibras multimodo es el tipo normalmente especificado para las LANs y WANs.

Cables de fibra Ópticas es costoso y requiere un profesional para su instalación y las conexión de dispositivos de redes. Con el crecimiento de del uso de fibra Ópticas, esta progresivamente requerirá menos niveles de experto ya que los avances tecnológicos están simplificando el proceso de instalación y conexión.

**Medio Wireless**

Redes de comunicación inalámbricas se han convertido más y más popular cada vez. Redes inalámbricas son comúnmente implementadas en ambientes híbridos. Un ambiente híbrido es aquel en el cual componentes inalámbricos se comunican con redes que utilizan cables. Por ejemplo, un computador tipo laptop puede usar sus capacidades wireless para comunicarse con la red corporativa LAN que utiliza cables para sus conexiones.

La única diferencia entre una LAN inalámbrica y una LAN cableada es sólo el medio, y su NIC y el transceptor para cada computador wireless. El transceiver es a menudo llamado un punto de acceso (access point) ya que este recibe y envía señales desde y hacia la red.

El uso de un medio de transmisión wireless para conectar una LAN tiene un uso muy selectivo en las arquitecturas de redes disponibles a la fecha. La necesidad de una LAN wireless y la existencia de una amplia gama de diferente tipos disponibles. Conectar edificios a larga distancia, lo que permite a los usuarios conexiones rápida, portable o temporal a una LAN o permitir acceso a una LAN móvil, encierra algunas de las razones para la capacidad de wireless en una LAN. Al considerar el uso de conexiones wireless en una LAN, existen tres métodos de transmisión que son Típicamente usados: transmisión de micro onda (microwave), transmisión infrarrojo (infrared), y la transmisión ondas de luz (lightwave).

***Transmisión Microwave***

Desarrollada por la industria telefónica para transferir llamadas de larga distancia, la tecnología microwave ha sido incorporada las operaciones de redes LAN. El espectro de frecuencias usado por las compañías telefónicas esta regido por el FCC (Federal Communications Commission) en los EEUU. Pero existe un conjunto de frecuencias por separado, como es el Industrial/Scie999999999 mi fie/Medical band, que no requiere licenciamiento del gobierno. Además, las frecuencias en el rango de 902- a- 928-MHz son frecuencias no restringidos. Por esta razón, LANs wireless locales que usan un medio de transmisión wireless pueden considerar este espectro de frecuencia para las redes de poca extensión.

La habilidad de conectar dos usuarios sobre una distancia amplia es posible en un sistema de transmisión de microwave. Típicamente sistemas de transmisiones microwave le permiten a los usuarios conectarse desde distancias hasta de 12 millas. Existen varias ventajas de la capacidad de relativamente larga distancias de conexión inalámbrica. Un uso común de este sistema de transmisión es para conectar edificios separados por distancia a la localidad Física de la LAN. Por lo general instalar conexiones físicas entre las dos localidades tienen un enorme costo y se puede tornar impractico. Si las dos localidades o los sistemas de transmisión están en una misma línea de visibilidad, la posibilidad de proveerle interconexión vía wireless LAN es posible.

Existen varias desventajas que tornan la transmisión de microwave impractica o imposible. Primer, para una conexión de larga distancia, las bandas de Industrial/Cientifica/Medical no pueden ser usadas como espectro de frecuencia de transmisión, y restricciones sobre la salida de potencia no permitirá las señales radiales viajar lo suficientemente lejos para completar una conexión a una distancia larga. Una transmisión típica de microwave es llevada acabo con frecuencias que operan alrededor de 10 GHz. Con estas frecuencias, proveído de que suficiente energía es transmitida con el equipo correcto, la transmisión de data a una LAN es posible. Eso si, este espectro de frecuencia requiere licenciamiento por las autoridades locales, en el caso de los EEUU, la FCC. Dependiendo del tiempo que la conexión necesita, obtener el licenciamiento para utilizar el espectro de frecuencia específico puede ser que se torne no práctico o mejor dicho rentable, para una conexión LAN de larga distancia puede tornarse como una opción no viable y que sea mejor conectarse a través del Internet y convertice en una extranet.

***Transmisión Infrarroja/Infrared***

Transmisiones Infrarrojas tienen utilidad en conexiones LAN modernas. El uso de esta tecnología es para comunicaciones a corta distancia. El uso primario de este espectro esta en dispositivos de control remoto para dispositivos multimedia como son televisores y VCR.

Existen muchos beneficios con el uso de tecnología de transmisión Infrarroja. El infrarrojo es relativamente direccional, económico, fácil de construir y mantener. La instalación de una red Infrarroja no requiere los cables utilizados en una LAN típica. Además, la versatilidad de los usuarios poderse conectar dinámicamente en una LAN requiere sólo confirmación vía software, previsto que tengan todo el equipo necesario para tener infrarrojo propiamente instalado en el LAN. Pero claro, existen desventajas primordiales en la tecnología de transmisión Infrarroja. Debido a la frecuencia de transmisión, las señales Infrarrojas no pueden traspasar objetos sólidos. Causado por esta limitación, las LANs Infrarrojas esta restringida a conexiones cortas y que no existan obstrucciones entre el transmisor y el recibidor. Otro limitante adicional es que transmisiones Infrarrojas deben estar libre de la luz del sol ya que este transmite mucha energía Infrarroja. Estar expuesta a la luz del sol para este tipo de LAN la tornaría sin uso.

El uso primario de conexiones Infrarrojas en redes LAN es conectar usuarios mó y temporales a la LAN. Eventos y Conferencias en las cuales computadoras portátiles serán usadas por corto tiempo pueden usar este tipo de transmisión para conectarse sin físicamente tener que conectarse a la LAN a través del uso de un trans-

misor localizado en el centro.

***Transmisión de lightwave (Onda de Luz)***

Muy a menudo es necesario proveer una localidad remota con la capacidad de accesar una LAN en la sede principal. La disponibilidad de poder proveer conexiones físicas con el uso de cables no siempre es posible, dependiendo de donde se encuentra el sitio remoto. Este problemas puede ser resuelto a través del uso de un sistema de transmisión de micro-ondas o uno de lightwave. Las limitaciones de sistemas de transmisiones microwave ya la explicamos; pero, un sistema de transmisión de lightwave tiene características similares que deben ser entendidas para así poder elegir entre el sistema apropiado de transmisiones wireless.

Un sistema de transmisión lightwave consisten en un láser y un foto detector, normalmente colocado en sima de edificios. Para colocar este tipo de conexiones, el perímetro de alcance de este sistema de transmisión es de varias millas. Aunque los sistemas de transmisión de ondas de luz funcionan bien en LANs que cubren corta distancia, existen algunas desventajas claves. Los rayos de luz tienen problemas al transmitir a través de precipitaciones atmosféricas, similar a la falta de visibilidad en los automóviles cuando se maneja a través de neblinas espesas. Condiciones climáticas adversas tornaran las conexiones a través de transmisiones vía ondas de luz inservible. Adicionalmente radiaciones solares reflejan desde edificaciones traspasan el rayo láser a aproximadamente la misma frecuencia e interfieren con las conexiones láser. Al ocurrir esto, el láser es refractado por el calor y falla los foto detectores.

***Satélite***

El uso de Satélite a lo que respecta a redes LAN de datos, aún se encuentra en pleno desarrollo. La disponibilidad de usar satélites para conectar a usuarios remotos a la red principal es el uso principal de este tipo de conexión. El uso de satélites para proveer conexiones LAN aún se tornan un poco difícil debido a unas cuantas desventajas.

Las instalaciones y los equipos que involucra sistemas de conexiones vía Satélite son costosos. La instalación del sistema requiere equipos especializados a un muy alto costo. Otra desventaja es la distancia que los datos tienen que viajar. Existe un retraso que no se puede obviar de hasta medio segundo al usar conexiones vía Satélite.

Causado por los costos aún extremos asociados al uso de conexiones satelitáles para las operaciones de redes LAN, la disponibilidad de este tipo de conexiones es relativamente bajo. Su uso primordial es la de conectar a usuarios individuales remotos, a las redes locales LAN de las compañías. Esto va cambiando rápidamente y con el crecimiento de las redes satelitáles puede ser que cambie en un tiempo prudente.

## TIPOS DE TRANSMISIÓN

Una vez establecida una red, los datos deben ser transmitidos por el medio elegido, en la mayoría de los casos es un cable. En esta sección discutimos algunos conceptos de transmisión de datos, que incluyen los siguientes:

- Transmisión Sincrónica
- Transmisión Asincroníca
- Flujo de Transmisión de Data
- Transmisiones Baseban y Broadband (Banda base y Banda Ancha)
- Topologías lógicas y Física

***Transmisión Sincrónica***

En transmisiones sincronizadas, el dispositivo de acceso y la red comparten el reloj y la velocidad de transmisión. A este tipo de transmisión se le llama sincronizada. Los datos son intercambiados a través de flujo de caracteres llamados tramas de mensajes de data (message-framed data). Una secuencia de inicio-y-pare (start-and-stop) es asociada con cada transmisión. Los dispositivos de acceso y de red necesariamente deben estar sincronizados para poder recibir el mensaje por completo en el orden que fué enviado o transmitido. Líneas T1 usan transmisiones sincronizadas.

***Transmisión Asincrónicas***

Transmisiones asincronizadas son caracterizadas por la ausencia de un reloj en el medio de transmisión. El dispositivo de acceso no está sincronizado con el dispositivo de red. Pero, las velocidades de transmisión deben ser la misma. así que los datos son transmitidos como caracteres individuales. Cada caracter es sincronizado por la información contenido en su bits de inicio (header, cabezal) y su pare (trailer, cola). Los tradicionales modems de conexiones Dial-Up usan transmisiones Asincrónicas.

***Flujo de Transmisión de Data***

Los tres métodos de operación de circuito son como sigue a continuación:

- **Simplex**
  Los datos viajan en una sola dirección, similar a un sistema de Aviso Público (Public Address, PA).
- **Half Duplex**
  Los datos viajan en dos direcciones pero solo en una dirección a la vez, similar a los walkie-talkie. El Ethernet usa transmisiones half-duplex.
- **Full Duplex**
  Los datos viajan en dos direcciones simultáneamente, similar a una conversación telefónica. Ethernet Full-duplex, es una extensión del Ethernet, soporta transmisiones full-duplex en un ambiente asistido por switch99999999999.

## Transmisiones de Banda Base y Banda Ancha (Basaban y Broadband)

En redes el bandwidth (ancho de banda) es la medida de la capacidad de transmisión de un medio en particular. Esta velocidad se cuantifica como un número de bits que puede ser transmitido por segundos. El ancho de banda de un medio de transmisión puede ser dividido en canales; así que cada canal es una porción de la capacidad total disponible para transmitir data. Los dos métodos usados para colocar ancho de banda a los canales son el Basaban y broadband.

***Basaban (Banda base)***

Basaban usa todo el ancho de banda del medio para un único canal. Aunque es usado principalmente para señales digital, Basaban puede también conducir señales análogas. La mayoría de las LANs, como son las redes Ethernet y Token Ring, usan señales digitales Basaban.

Basaban usan un tecnología de transmisión llamada TDM (Time División Multiplexing, TDM). Esta tecnología TDM envía multiple señales sobre una sola ruta de transmisión y lo logra tejiendo la señales. Por ejemplo, tres señales (1, 2, y 3) pueden ser enviadas así 112233112233. El dispositivo que recibe separa este flujo único en las tres señales originales. TDM Estatistico99999 (StatTDM) le da más prioridad a señales más urgentes.

***Broadband***

Broadband divide el ancho de banda del medio en múltiples canales, y cada canal transmite una señal por separado. Este método permite que un singular medio de transmisión trasmita varias conversaciones simultáneamente y sin ninguna interferencia. El método broadband es usado exclusivamente para señales análogas; este no puede ser aplicado a señales digital porque estas señales pueden interferirse entre ellas. así que, transmisiones broadband requieren que un modem le convierta señales digitales a análoga y vice versa.

El método broadband usa un tecnología llamada FDM (Frequency División Multiplexing). Al igual que TDM, FDM también transmite multiple señales sobre una sola ruta de transmisión. Pero, cada señal en FDM transmite en un rango de frecuencia única, o carrier. Transmisiones broadband son muy comúnmente usadas para TeleCable y sistemas inalámbricos o wireless.

El término broadband es comúnmente usado para describir cualquier transmisión de data de alta velocidad que proveé servicios a velocidad de T1 (1.544 Mbps) y más alta. Eso si, las capacidades de la tecnología broadband varia inmensamente dependiendo de la situación, y velocidad actual de transmisión puede que no logren los niveles de Tl o tal ves sobre pasarlos por mucho. Por lo general, broadband implica velocidades de transmisión muy superior a aquellas que estaban disponibles en el pasado.

**Topologías lógicas y Física**

Topologías lógicas se refiere a la ruta real sobre una red de una señal generada. Bus y Ring son dos tipos de topologías lógicas. Una red lógica del tipo bus genera una señal a todos los dispositivos en una red. Una red lógica del tipo ring genera una señal que viaja en una dirección por una ruta determinada. En este capítulo discutiremos a más profundidad redes lógicas introduciendo sus métodos de acceso.

Topologías físicas se refiere a la manera en que los dispositivos de una red están conectados. Ya hemos vistos topologias físicas cuando estudiamos topologias bus, star, ring, y mesh, anteriormente en este mismo capítulo. Es muy importante entender la diferencia entre topologias lógicas y físicas.

**DATA LINK LAYER (CAPA de ENLACE de DATOS)**

En esta sección estudiamos la segunda capa de la pila OSI, la capa de Enlace de Datos. Analizamos direccionamiento, controlando y sincronizando flujo de data y revisamos errores y lo corregimos.

La capa de Enlace de Data maneja:

- **Data addressing**
  Direccionamiento es la asocia de una dirección con cada host del lado de la capa Física, similar a una dirección postal.
- **Flow control**
  Control de Flujo, controla flujo de data sobre la capa Física; Ej., ¿qué pasa si dos hosts desean enviar data simultáneamente? (Este es mejor conocido como Control de Acceso al Medio.)
- **Data integrity**
  Integridad de Data se asegura que la data llega a su destino intacta y en el orden correcto- revisa por errores.

Estos tres requerimientos son esenciales para cualquier comunicación confiable sobre un medio físico.

Los siguientes tópicos son discutidos en la siguiente sección:

- Direccionamiento de Data (Data Addressing)
- Control de Flujo (Flow Control)
- Integridad de Data (Data Integrity)
- Tramas (Frames)
- Acceso a la Capa Física

**Direccionamiento de Data**

Las direcciones de la capa de Enlace de Data deben ser únicas en toda la capa Física (Ej., sobre una red local). De otra forma, tráfico de la red no puede ser direccionado a su destino correcto.

***Direcciones MAC***

Cada dispositivo en una red debe tener una dirección única. Los usuarios pueden usar nombres como Oficina, pero la capa de Enlace de data usa completamente un sistema numérico de direcciones. Este es de 48 bits (6 bytes) de largo, con 3 bytes reservadas para el ID del fabricante y 3 para uso local. Esta direc-

ción, el número MAC (Media Access Control), es asignada por el protocolo ARP (Address Resolution Protocol). Direcciones MAC son direcciones únicas que son parte del NIC. Cada dirección es asignada por el fabricante y es usada para identificar la computadora en la red.

Direcciones MAC son llamadas direcciones físicas y no lógicas. Direcciones lógicas son encontradas en la Capa 3 (Capa de Red) y incluyen direcciones IP y IPX. Direcciones lógica son usadas para enviar data sobre el Internet y la red enviadas a destinaciones remota. Direcciones físicas son encontradas en la Capa 2 (Capa de Enlace de Data) y son muy a menudo parte de la interfase Física. Direcciones Física son usadas solamente para enviar data entre dos dispositivos en un sólo enlace de red.

Direcciones MAC usan un hexadecimal de 12 dígitos para formar una dirección de 48-bit (6 bytes). Cada mitad de la dirección es usada para un propósito diferente.

| | | |
|---|---|---|
| **00-0A-95** | **-** | **FF-FE-85** |
| **Código del Fabricante** | | **número Serial del Interfase** |

El código del fabricante está identificado en los primeros 24 bits (3 bytes).

El número serial de la interfase está identificado en los últimos 24 bits. Determinado por el fabricante , es siempre único para ese fabricante. En teoría, ninguno dos direcciones MAC son idénticas.

***Determinar direcciones MAC en GNU/Linux***

Usted puede determinar la dirección MAC de su sistema usando el comando ifconfig (interfase configuration) con la opción -a (deberá estar ingresado como root y la interfase deberá estar activa para que se pueda desplegar la dirección MAC).

Aquí un ejemplo de la salida del comando ifconfig:

```
# ifconfig
```

Resultados del comando ifconfig, dependiendo de su NIC y la configuración de su red:

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:08:C7:F3:E6:E4
          inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
          EtherTalk Phase 2 addr:65280/106
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1179092 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1583178 errors:1 dropped:0 overruns:0 carrier:1
          collisions:0
          RX bytes:208032912 (198.3 Mb)  TX bytes:1540098887 (1468.7 Mb)
```

### Ejercicio 2-1: Identifique la Dirección de Hardware en GNU/Linux

En este ejercicio, usted va a localizar la dirección del hardware de su computador en GNU/Linux. No se proveen soluciones a este ejercicio.

1.- Ingrese al sistema como root.

2.- El prompt digite:

```
# ifconfig
```

3.- Aparecerá la información de su IP y la información de su NIC. Observe la dirección de su hardware (HWaddr).

### Address Resolution Protocol (ARP)

El protocolo de Resolución de Direcciones para mapear direcciones IP a direcciones de hardware, una variedad de opciones pueden ser usadas. ARP es usado Típicamente en redes IEEE 802 (Ethernet, Token Ring). Este proveé una vía para que un sistema difunda una petición para una dirección IP para una dirección de hardware. Los broadcast de ARP a todos los hosts en una red local. La información puede ser proveída por el host en específico o por otros hosts que tal ves ya saben la repuesta. REF 826 cubre la conversión de direcciones IP a direcciones MAC.

Aquí presentamos un ejemplo de como funciona la resolución con ARP:

- Cuando el host cliente1 necesita resolver la dirección del cliente2, este envía un broadcast de un paquete especial que pide a cliente2 que responda con su dirección Física. Este mensaje es conocido como el paquete de petición de ARP.
- Aunque todos los hosts en la red reciben la petición, sólo el nodo que reconoce su dirección de Internet responde con su dirección Física. este mesase es referido como la respuesta ARP.

Los hosts que usan ARP mantienen un cache de las direcciones adquiridas recientemente de Internet-a-Física para así ellas no tener que usar ARP repetidamente. El tiempo en averaje que una dirección ARP permanece en un cache GNU/Linux es de 20 minutos.

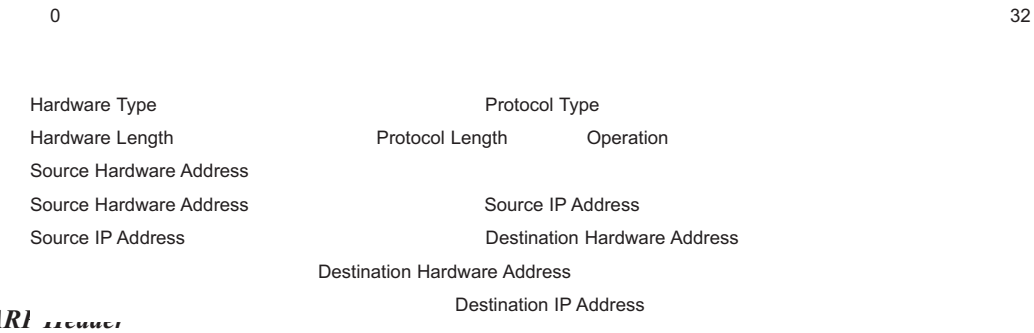
El comando para ver el cache del ARP en GNU/Linux es:
**# arp**

Este comando genera el siguiente resultado:

EJEMPLO DEL COMANDO ARP 99999999999

**Cabezal del ARP (Header)**

El ARP header tiene una longitud de 28 bytes. El formato del header consiste de palabras de 32-bit. Este formato es usado para el propósito de desplegar para así poder entenderlo y explicarlo. El header del ARP consiste de siete palabras de 32-bit, lo cual es 28 bytes.




***Campos de ARP Header***

Un mensaje ARP es encapsulado en una trama de Ethernet. El campo frame type del header de Ethernet para los paquetes ARP se establece al hexadecimal 0806. Los siguientes son los campos del header:

**Hardware Type (16 bits)**
Define el tipo de dirección de Hardware (uno para Ethernet)

**Protocol Type (16 bits)**
Define el tipo de dirección del protocolo (0x0800 para las direcciones IP); el mismo valor que el campo tipo de frame de Ethernet.

**Hardware Length (8 bits)**
Tamaño, en bytes, de la dirección de hardware [el valor es de 6 bytes (48 bits) para Ethernet]

**Protocol Length (8 bits)**
Tamaño, en bytes, de la dirección del protocolo [el valor es de 4 bytes (32 bits) para el IP]

**Operation (16 bits)**

Define el tipo de ARP: 1=Peticiones de ARP 2=Respuestas de ARP

**Source Hardware Address (32 bits)**
Dirección de hardware del que envía (6 bytes para Ethernet)

**Source Hardware Address (16 bits)**
Dirección de hardware del que envía (continuado)

**Source IP Address (16 bits)**
Dirección del protocolo de equipo que envía (4 bytes para el IP)

**Source IP Address (16 bits)**
Dirección del protocolo de equipo que envía (continuado)

**Destination Hardware Address (l6bits)**
Dirección de hardware de equipo de destino (6 bytes para Ethernet)

**Destination Hardware Address (32 bits)**
Dirección de hardware de equipo de destino (continuado)

**Destination IP Address (32 bits)**
Dirección del protocolo de equipo que envía (4 bytes para el IP)

### Ejercicio 2-2: Ver el Cache del ARP en GNU/Linux

En este ejercicio, echaremos un vistazo al cache del ARP, entonces procederemos a agregar y eliminar entradas en el cache. No se proveen soluciones para este ejercicio.

1. En el prompt de root (#), escriba el siguiente comando:
   **# arp**

2. Visualice las entradas del cache de ARP. Si una entrada no existe de su segundo equipo, creala ejecutando este comando en el prompt:
   **# ping [dirección IP de segundo sistema no existe]**

3. Presione las teclas CONTROL+C para detener el proceso de ping. Visualice de nuevo el cache de ARP, escribiendo este comando:
   **# arp**

   Ya debe existir una entrada para su segun sistema en el cache de ARP.

4. escriba la entra ARP de su segundo sistema. Incluya solamente las direcciones IP y de hardware:

   Dirección IP: _____

   Dirección de HW:_____

5. Si existen entradas adicionales en su cache de ARP, intente determinar a que nodos de la red pertenecen- Por ejemplo se ha usted comunicado recientemente con otro equipo en la red?

6. Para eliminar una entrada de ARP, use la opción -d. En el prompt simplemente escriba:
   **# arp -d [Dirección IP segundo equipo en el sistema]**

7. Visualice el cache del ARP escribiendo:
   **# arp**

   Note como el ARP de su segundo sistema ya no aparece como entrada en el cache ARP.

### Protocolo de Resolución de Direcciones Inverso (RARP)

El RARP (Reverse Address Resolution Protocol) es usado por los sistemas sin discos para poder encontrar su dirección de Internet en la red. El sistema sin disco (diskless system) efectúa una petición RARP a través de un broadcast, el cual proveé su dirección Física en la red. Un servidor RARP entonces procede a enviar un RARP de repuesta, normalmente un unicast, el cual específica la dirección IP de la estación

sin discos.

**Ethernet Address (48-bit) ===== > IP Address (32-bit)**

Generalmente los sistemas sin discos dependen de RARP durante su inicialización. Soporte de RARP puede ser proveído en ROM ya que es pequeño y simple. Debe existir por lo menos un servidor RARP en la red para que RAPR funcione.

**Descripción de RARP**

El header RARP tiene la misma longitud como el header ARP (28 bytes) y también es encapsulado en la porción de la trama Ethernet que pertenece al campo de Data. Este permite no sólo a que las computadores adquieran su dirección de Internet, sino que también esas de los otros sistemas. Todo computador en la red recibe una petición, pero sólo el servidor RARP procesa la petición y envía la respuesta.

Los headers del RARP y el ARP son similares. Sus únicas diferencias son ilustradas a continuación:

• Los paquetes de petición y respuestas RARP tienen su encabezado de Ethernet el campo tipo de trama establecido al hexadecimal 8035.
• El campo de operación define los tipos de mensajes RARP:
3=RARP request(petición)
4=RARP reply (`respuesta)

**Control de Flujo (Flow Control)**

Si A envía datos a B más rápido que B puede procesarlo, B se sobrecarga. El Control de flujo se implementa estableciendo el número máximo de tramas que A puede enviarle a la ves a A antes de que B le permita continuar.

Estas medidas son implementadas por lo general en las capas superior de los protocolos pero también puede ser implementado en la capa de Enlace de Data. En el caso del Ethernet y el TCP/IP, la capa de Transporte (TCP) maneja el control de flujo donde sea necesario aplicarlo. Hay un mecanismo para manejar la sobrecarga de los hosts llamado Source Quench. Este mecanismo yace entre el ICMP (Internet Control Message Protocol), un protocolo de la capa de Red que maneja el control de flujo para el IP. El mecanismo envía un mensaje al dispositivo que envía (origen) para detectar su velocidad de envío. Este es reactivo y no premeditado. Hay una gran variedad de mensajes ICMP, así como es determinar si un host esta despierto o no, petición de eco (echo), parámetro equivocado en un header de IP, o redireccionar mensajes por otra ruta.

El protocolo de alto nivel HDLC (High-level Data Link Control) es un ejemplo de control de flujo en esta capa. Es usado principalmente por paquetes provenientes de sistemas que proveen switching a paquetes generados por el teléfonos como son los protocolos X.25 o el Point-to-Point Protocol (el PPP, será discutido más adelante).

**Integridad de Data**

La integridad de data es proveída con la división de la data en tramas y agregandole un CRC (Cyclic Redundancy Check). El CRC es un código computacional diseñado matemáticamente para servir como una firma electrónica para trama de la data involucrada (Header + Data).

**Data Frame ---> | HEADER | DATA | CRC |**

Si aunque sea un sólo bit de data dentro de la trama se corrompiese dentro del frame (por ejemplo, interferencia eléctrica), entonces cuando la data es recibida en el destino un nuevo CRC es computado desde la trama de datos recibida, el nuevo CRC variará comparado con el transmitido originalmente en la trama. así que, el host de destino sabrá que la trama está corrompida y entonces decidirá que otra acción tomar. En

muchas tecnologías LAN, la acción simplemente es dejar caer la trama y olvidarse que nunca existió sin notificar al enviante o originador.

La matemática detrás de los CRCs ha sido cuidadosamente diseñada para reducir el chance de que tramas dañadas generen el mismo CRC que el original sea virtualmente cero. así que, previsto que el CRC calculado una vez recibido el frame iguala el enviado, entonces se asume que la trama está intacta.

**Timers y Timeouts**

El revisado del CRC se asegura que tramas o frames corrompid sean aceptadas por el host de destino. Pero, si es necesario que una trama corrompida sea devuelta al host de origen. ¿Como se lleva esto acabo?

El host A establece un tiempo al momento de enviar la trama. Si el, el host A, no recibe una confirmación (acknowledgment) a tiempo, la trama es reenviada y el timer es reiniciado.

Mientras que esto es una característica de la capa de Enlace de Data, no es implementado allí (no en redes de Ethernet), y esta responsabilidad de la verificación del transporte de los datos es dejado a capas más altas del OSI. Con el TCP/IP sobre Ethernet, la capa de Red (con el IP, también sucede lo mismo veremos después) es quien maneja esto.

Este proceso requiere que cada trama sea enumerada para poder llevar acabo su confirmaciones y que sus tramas reconocidas puedan ser identificadas.

**Tramas/Frames**

Es necesario que los datos que colocamos en un alambre sean particionados en pequeños paquetes discretos de información. Estos paquetes son denominados tramas en la capa de Enlace de Data. Existe un gran número de tramas que podemos usar que varían por su formato, tamaño, codificación y todos los parámetros del protocolo que pueden ser usados. Diseñar un nuevo protocolo de Enlace de Data no es una tarea fácil! Aquí mencionamos algunas de las cosas que se deben tomar en cuenta:

- ¿Cómo son las tramas delimitadas?
- ¿Qué tamaño deben ser las tramas?
- ¿Cómo deben ser los bits codificados en el medio físico?
- ¿Qué sucede si dos computadoras intentan enviar data a la misma vez?
- ¿Cómo sabe usted si la trama llegó a su destino?
- ¿Cómo controla usted a un host de no sobrecargar el otro con el envío de su datos?
- ¿Cómo podemos optimizar todo este proceso?

Hay tres maneras diferentes de marcar el la longitud de una trama y como mantenerlas separadas. Podemos marcar el final de una trama con delimitador especial. También podemos colocar el largo de la trama en el cabezal (header) de la trama misma. Otra posibilidad es que todas las tramas sean de la misma longitud y enviarlas de acuerdo a un mecanismo especial de el timing. La mayoría de medio físicos usan el método de colocar la longitud de sus tramas en el header.

**End marking: | START | HEADER | DATA | CRC | END|** ***----Marcadas al final***
**Length mark: | HEADER | LENGTH | DATA | CRC |** ***---Marcadas su longitud***
**Timing: | HEADER | DATA | CRC | .....pausa después de cada trama. *---Un timer interno***

Un preámbulo es generalmente agregado al principio de cada trama para permitirle a la tarjeta de interfase que sincronice su reloj con la trama que llegan

.

**| 101010010...01001001...010101010 | TRAMA |**

***Tamaño de la Trama***

El tamaño máximo de un frame en una red, determina la carga máxima de datos que cada frame puede cargar a la vez. Esta carga máxima es conocida como el MTU (Maximum Transmission Unit) del sistema. Si un host desea enviar pedazos de data más grande que el MTU, esta deberá repartirse en más de un trama. Consideremos lo siguiente:

- ¿De que tamaño debe ser la trama?
- Los headers y los CRCs son por lo general de un tamaño fijo.
- Tramas con tamaño variables son usadas con un tamaño permisible máximo o mínimo.

## Codificación Manchester

Hay mucha maneras de codificar una cadena de datos a voltajes eléctricos en una línea serial. Una manera común es establecer cada estación en una frecuencia en particular, digamos 9600 baudios, y traducimos un bit 0 a 0 voltios y el bit 1 al voltio de la línea. Entonces rodeamos cada byte con un start, stop, y bits de paridad para sincronizar y revisar la información. En la codificación Manchester, usamos la transición de voltaje y no el voltaje simple. Otro método es la codificación Manchester diferencial que consiste en que un bit con valor 1 se indica por la ausencia de transición al inicio del intervalo, y un bit con valor cero se indica por la presencia de una transición al inicio del intervalo. En ambos casos, existe una transición en la parte media del intervalo.

## Acceso a la Capa Física

Necesitamos un método para controlar el acceso a la capa Física. Si todos los ordenadores intentan enviar datos a la vez, se perderían en colisiones. Existen dos maneras esenciales de enfrentar este dilema: Carrier Sense Multiple Access/Collision Detection (CSMA/CD) y paso de ficha (token passing).

Las colisiones resultan en perdida de data. Los CRCs no se ajustan a la reconstrucción de data dañado o perdida en una trama, así que no se efectúan intentos de salvar los resultados de una colisión. Las colisiones pueden ser detectadas fácilmente, pero, por host que continuamente monitorean el medio por otras data mientras sigue enviando.

***CSMA/CD***

El protocolo CSMA-CD funciona de la siguiente manera: un nodo que desea transmitir espera a que el canal esté aislado, escuchando por períodos de silencio antes de enviar sus tramas (CS), una vez que se encuentra en este estado empieza la transmisión. Las tramas toman un tiempo determinado para viajar por el medio. Si el nodo A empezara también a transmitir mientras B aún piensa que la línea está desocupada, en este instante se produciría colisión, por lo tanto se detiene la transmisión y se retransmite tras un retraso aleatorio.

Esto es muy similar a conversaciones de dos o más personas que se interrumpen comenzando a hablar a la misma vez. Dos computadores pueden determinar en un mismo momento que el cable esta libre.

Todo esto es manejado con el CSMA/CD, de esta manera:

- Los Hosts pueden detectar cuando una colisión está ocurriendo y parar de enviar (CD).
- Ellos entonces esperan un tiempo al azar antes de intentar enviar de nuevo si la línea ya esta limpia.
- Reintentos futuros pueden ser colocados más atrás aún en tiempo exponencial

  Ejemplo es si se falla la segunda vez es dos veces más, entonces 4 veces más, 8 veces, 16 veces, etc.

Las variaciones de CSMA/CD son denotadas por su persistencia, esto significa que, aunque ellos traten siempre de enviar inmediatamente después que la línea este limpia.

***CSMA 1-persistente***

El protocolo CSMA 1-persistente funciona de la siguiente forma: cuando tiene que transmitir un frame, primero escucha el canal y si está libre envía el frame, caso contrario, espera a que se libere y en ese momento lo envía. Se denomina CSMA 1-persistente porque existe la probabilidad 1, es decir, certeza de que el frame se transmitirá cuando el canal esté libre. En una situación real con alto tráfico es muy posible que cuando un nodo termine de transmitir existan varios esperando enviar sus datos, y con CSMA 1-persistente todas los frames serán emitidos a la vez y colisionarán, pudiéndose repetir el proceso varias veces con la consiguiente degradación del rendimiento. Cabe señalar que una colisión ocurrirá aunque no empiecen a transmitir exactamente a la vez, basta simplemente con que dos nodos empiecen a transmitir con una diferencia de tiempos menor que la distancia que los separa, ya que en tal caso ambos detectarán el canal libre en el momento de iniciar la transmisión. Se deduce entonces, que en este tipo de redes el retardo de propagación de la señal puede tener un efecto importante en el rendimiento. El rendimiento obtenido con este protocolo puede llegar al 55% con un grado de ocupación del 100%.

El Ethernet usa el CSMA/CD 1-persistente, significando que esta siempre trata de enviar las tramas a respuesta de una línea vacía (esto significa que, la probabilidad de transmisión es 1).

***CSMA: no persistente***

Corresponde a una modificación del protocolo anterior que funciona de la siguiente manera: antes de enviar se escucha el canal, si el canal está libre se transmite el frame. Si está ocupado, en vez de quedar escuchando, se espera un tiempo aleatorio, que viene dado por un algoritmo llamado de backoff, después del cual se repite el proceso. El protocolo tiene una menor eficiencia que CSMA 1-persistente para tráficos moderados, pues introduce una mayor lactancia; sin embargo se comporta mejor en situaciones de tráfico intenso ya que evita las colisiones producidas por las estaciones que se encuentran a la espera de que termine la transmisión de un frame en un momento dado.

***CSMA p-persistente***

Este protocolo utiliza intervalos de tiempo y funciona de la siguiente manera: cuando el nodo tiene algo que enviar primero escucha el canal, si está ocupado espera un tiempo aleatorio. Cuando el canal está libre se selecciona un número aleatorio con distribución uniforme entre 0 y 1, si el número seleccionado es menor que p el frame es transmitido. En caso contrario, se espera el siguiente slot de tiempo para transmitir y repite el algoritmo hasta que el frame es transmitido o bien otro nodo utiliza el canal, en cuyo caso se espera un tiempo aleatorio y empieza de nuevo el proceso desde el principio. La eficiencia del protocolo es, en general, superior a la de CSMA 1-persistente y CSMA no persistente.

Entre las Ventajas CSMA/CD se incluyen:
- Simplicidad del algoritmo
- Fácil incluir y eliminar nuevos hosts
- Implementación sencilla, bajo coste y fiabilidad
- Requiere poca administración
- Técnica suficientemente probada
- Buen manejo de carga de variables
- Buen rendimiento hasta determinado nivel de carga (aproximadamente 85%)

Inconvenientes CSMA/CD incluyen:
- Técnica LIFO ante colisiones
- Longitud de los mensajes mínima

• Difícil distinción entre ruido y colisiones
• La atenuación complica la detección de colisión
• No permite la gestión de prioridades
• No garantiza tiempo de entrega
• Difícil ejecutarse a alta velocidad
• Rendimiento pobre conforme aumenta la carga (colisiones masivas y retrasos)

Aún con estas desventajas, el CSMA/CD ayuda a hacer que el Ethernet sea la elección más popular en networks al día de hoy causado por su simplicidad de implementación y el bajo coste de su hardware.

***Token Passing***

Este protocolo, que se utiliza en redes Arcnet y Token Ring, se basa en un esquema libre de colisiones, dado que la señal (token) se pasa de un nodo o estación al siguiente nodo en la anillo. Con esto se garantiza que cada estación solamente envía tramas mientras posé el token y que un sólo paquete viajará a la vez en la red lo que significa que no habrán colisiones.

Claro está, no colisiones no significa que no hay problemas. Por ejemplo, ¿que sucede si se corrompe el token mientras se pasa alrededor del ring?

Cada host tiene que saber cual máquina están en cada lado de su anillo lógico para así poder aceptar y pasar el token. Considere que sucede si agregamos un host extra a la red- no esta dentro del anillo y por esto no recibirán automáticamente el token.

Resolver este y otros problemas hacen de pasar el token (aunque simple a primera vista) un sistema mucho más complejo de implementar que CSMA/CD.

La configuración anterior es conocida como token bus. La de Token Ring es muy similar, pero los hosts son ambos lógicos y físicamente conectados al ring.

Entre las ventajas de token passing se incluyen:
• Buen manejo de carga variable
• Sin Colisiones
• Garantía de tiempo de entrega/mínimo uso ancho de banda
• Maneja bien sobrecarga
• Puede ser ejecute a alta velocidad

Entre las desventajas de token passing se incluyen:
• Agregar y eliminar hosts no es simple tarea.
• Si un host falla, la red se apaga.
• En practica, es compleja de implementar y administrar.
• Por lo general, altamente compleja significa alto coste.

Un sistema token-passing es más eficaz en un ambiente donde la data debe tener una garantía de tiempo de entrega o una disponibilidad de ancho de banda mínimo. Cuando el nivel de ejecución es lo más importante, o la red es probable que este muy sobre cargada, el token passing es lo ideal.

## ESTÁNDARES DE COMUNICACIÓN

Un factor que distingue las tecnologías LAN es sus métodos de acceso. Los métodos de acceso se refiere a la manera que los datos son colocados en el alambre físico. La serie 802 incluye CSMA/CD, token, y el método de acceso de prioridad en demanda (demand priority access methods). Cada uno de estos

métodos será discutido en esta sección.

En esta sección, los siguientes tópicos serán discutidos:

- Estándares LAN
- Estándares WAN

| Comparativa de Tecnologias LAN | | | | | |
|---|---|---|---|---|---|
| | **10Base2 Ethernet** | **10Base5 Ethernet** | **10BaseT Ethernet** | **100BaseT Ethernet** | **IBM Token Ring** |
| **Medium** | Coaxial | Thick Coaxial | Twisted Pair | Twisted Pair/Fiber | Twisted Pair/Fiber |
| **Topology** | Bus | Bus | Star | Star | Ring |
| **Max segments length** | 185m | 500m | 108m | 100m | Max 260 hosts |
| **Max nodes/segment** | 30 | 100 | 1 | 1 | N/A |
| **Connectors** | BNC | 50/8-pin AUI | RJ-45 | RJ-45/Photodiode | 9-pin/RJ-45 |
| **Speed** | 10Mbps | 10Mbps | 10Mbps | 100Mbps | 4 o 16 Mbps |

- Serial Line Internet Protocol (SLIP)
- Protocolo de Punto-a-Punto (Point-to-Point Protocol, PPP)
- Redes Digital de Servicios Integrados (Integrated Services Digital Network (ISDN)
- Cable Modem
- Digital Subscriber Line (DSL)
- X.25
- Conmutación rápida de paquetes/Fast Packet Switching
- Sistemas de T-Carrier
- Sistema de E-Carrier

## Estándares LAN

El Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers, IEEE) es una organización de profesionales que se encarga de crear estándares para computadores y las comunicaciones. La serie de Estándares IEEE 802 especifican varias tecnologías LAN. Refierase a las páginas web de la IEEE en http://www.ieee.org para más información.

En el ambiente TCP/I Ethernet es el LAN más comúnmente usado. La arquitectura lógica es de muchos sistemas diferentes conectados a un cable central o bus, llamado el ether. De esta manera, cada nodo puede potencialmente conectarse con cualquier otro nodo.

Otra tecnología dominante para LANs basadas en arquitecturas de PC es el Token Ring, más trabajado por IBM. Se fundamenta en que un token circula alrededor de todo el añillo todo el tiempo. Si una estación de trabajo desea enviar data, deberá adquirir el token y enviar su mensaje adjunto a este. Esto se asegura de que sólo una estación transmitirá mensajes a la vez.

Aquí le presentamos una comparativa de varias tecnologías LAN comunes.

Esta sección cubre los siguientes Estándares de red IEEE 802:

- IEEE 802.2
- IEEE 802.3-Ethernet
- IEEE 802.3u-Fast Ethernet

- IEEE 802.3z y el 802.3ab-Gigabit Ethernet
- IEEE 802.5-Token Ring
- IEEE 802.12-100VG-AnyLAN

En esta sección discutiremos dos Estándares de LAN que no están incluidos en los Estándares IEEE. Estos Estándares son o propietarios o estandarizados por otra organización. Ellos son:

- Apple LocalTalk
- Fiber Distributed Data Interfase (FDDI)

**IEEE 802.2**

Todos los Estándares en la serie IEEE 802 usan el estándar 802.2. Este estándar también es usado por redes de Fiber Distributed Data Interfase (Interfase de Data Distribuida en Fibra, FDDI), el cual no es un estándar de la IEEE. El estándar 802.2 divide la capa de Enlace de Data del modelo OSI en dos subcapas: Control de Enlace lógico (Logical Link Control, LLC) y Control de Acceso al Medio (Media Access Control, MAC).

La capa de la subcapa LLC proveé servicios orientados a conexiones y servicios sin conexión en la capa de Enlace de Data, cual administra transmisiones y puede proveer control de flujo.

La subcapa de MAC proveé:

- Acceso al medio del LAN. La MAC es responsable por colocar la data en el alambre.
- La dirección MAC (también llamada la dirección de hardware, dirección física o dirección de Ethernet).

Aunque Ethernet no usa técnicamente el estándar 802.2, este es compatible con el y comparte varios elementos importantes, como es la subcapa MAC.

**IEEE 802.3-Ethernet**

Ethernet es sin lugar a duda una de las tecnologías LAN más exitosas y es un predecesor al estándar IEEE 802.3. Es un sistema de transmisión para la comunicación entre sistemas. Este utiliza los Estándares de alambrado 10Base2, 10Base5, o 10BaseT. También puede usar cables de fibra óptica.

Aunque Ethernet y IEEE 802.3 son ambos soportados y usados intercambiables, Ethernet no conforma totalmente con el estándar 802.2.

La diferencia entre IEEE 802.3 y Ethernet no afecta a los fabricantes de hardware porque ambos el IEEE 802.3 y el Ethernet soportan direcciones MAC y la misma capa Física. Además, existe software que diferencian entre las subcapas.

Todas las redes que usan Ethernet/IEEE 802.3 (incluyendo IEEE 802.3u, 802.3z, y el 802.3ab) usan CSMA/CD. Una estación de trabajo debe asegurarse que ninguna otra comunicación se encuentra ya desarrollandose. Si ninguna otra estación se encuentra transmitiendo, el que envía puede empezar de inmediato. Las colisiones ocurren cuando dos o más estaciones determinan que el canal esta libre y empiezan a transmitir simultáneamente. En caso de una colisión, toda la transmisión cesa, mientras las estaciones en conflictos son notificadas. Las estaciones que reportaron colisiones entonces esperan un tiempo al azar antes de transmitir de nuevo.

Las transmisiones son locuciones a todas las estaciones de trabajo. Solamente el sistema de destino responde; todos los otros sistemas descartan la transmisión. Este proceso puede crear tráfico pesado en una red. así que , es importante dividir grandes redes de ethernet en segmentos (usando un bridge).

**IEEE 802.3u- fast Ethernet**

El Fast Ethernet es una versión más rápida del IEEE 802.3. Fué desarrollada originalmente por fabricantes como son 3Com, Intel, Cabletron, SynOptics, Digital, Grand Junction Networks. El comité del IEEE 802.3 es responsable por el Fast Ethernet. El objetivo principal del estándar Fast Ethernet es promover el uso del Ethernet de 100 Mbps usando el mismo método de acceso, CSMA/CD.

Fast Ethernet soporta los Estándares de cableado de 100BaseTX y 100BaseT4, cuales requieren cablea-

| Descripción | Ethernet | Fast Ethernet |
|---|---|---|
| Speed | 10 Mbps | 100 Mbps |
| IEEE Standard | IEEE 802.3 | IEEE 802.3u |
| Access Method | CSMA/CD | CSMA/CD |
| Topology | Bus/Star | Star |
| Cable Support | Coax/twisted Pair/fiber | Twisted Pair/Fiber |
| UTP Link Distance (Maximum) | 100 meters; | 100 meters; |

do UTP Categoría 5 para soportar los 100 Mbps. Este puede también usar 100BaseFX, cual es el cableado de fibra óptica. Fabricantes soportan tarjetas nics Fast Ethernet que usan ambas velocidades de transferencia 10 Mbps y 100 Mbps.

La mayoría de administradores de redes ya se encuentran actualizados a desde las 10BaseT a las 100BaseTX o 100BaseT4. En muchos otros casos encontraras que esta actualización fué efectuada con el reemplazo de las NICs 10BaseT con las 100BaseTX o las NICs 100BaseT4 y actualizando los HUBS para soportar ambas la 10BaseT y 100BaseTX o las 100BaseT4. Este proceso puede ser menos costoso que el de actualizar a una NIC 100BaseFX, Token Ring de 16 Mbps, 100VG-AnyLAN, o redes FDDI.

En la siguiente tabla mostramos las diferencias entre Ethernet y el Fast Ethernet:

**IEEE 802.3z y 802.3 ab-Gigabit Ethernet**

El Gigabit Ethernet es la tecnología LAN más veloz disponible 802.3. Es usada principalmente en los backbones de las redes corporativas. El estándar de Gigabit Ethernet transfiere data a velocidades hasta los 1,000 Mbps usando el método de acceso CSMA/CD.

Los dos tipos de Gigabit Ethernet son IEEE 802.3z y 802.3ab. El estándar 802.3z es especificado para cables especiales de cobre y cables de fibra óptica. El estándar 802.3ab específica el Gigabit Ethernet sobre cable UTP Categoría 5 UTP (hoy día ya se llego a la Categoría 6 de Cables UTP).

Gigabit Ethernet soporta el estándar de cableado de 1000BaseT, el cual usa cableado UTP Categoría 5 para poder soportar los 1,000 Mbps. Este también puede usar 1000BaseCX, 1000BaseSX, y 1000BaseLX , cual usa cableado de fibra óptica.

**IEEE 802.5-Token Ring**

Las red Token Ring está especificada en la definición IEEE 802.5. Token Ring fué originalmente desarrollada por IBM para su ambiente de mainframe, y el estándar IEEE 802.5 cumple con el desarrollo original de la corporación.

Mientras que Ethernet usa el método de acceso CSMA/CD. Redes Token Ring usan el método de acceso de token-passing (paso de ficha) que discutimos anteriormente. En vez enviar broadcasts como hace el Ethernet, una red Token Ring pasa un token en una dirección alrededor de la red. Cada nodo procesa el token

para determinar su destino. El después de revisar o lo acepta o lo coloca en el anillo de la red. Uno más tokens pueden circular el anillo (ring). Con el paso de ficha, no ocurren colisiones; es muy similar a calles de una sola vía sin esquinas que la crucen.

El estándar IEEE no específica un estándar de cableado, pero las redes de Token Ring de IBM usan cables twisted-pair. velocidad de transferencia de 4 o 16 Mbps son posible con redes Token Ring. Redes Token Ring aparentan estar usando topología de red STAR cuando en realidad usan un dispositivo algo parecido a un HUB llamado MSAU (Multistation Access Unit) para formar el anillo.

El MSAU crea el anillo usando conexiones internas.

El fallo de un nodo en la topología anilla RING puede causar la caída de la red por completo. así que, los MSAUs pueden identificar y sobre volar nodos fallidos en un segmento de red y continuar funcionando.

**IEEE 802.12-100VG-AnyLAN**

La 100VG-AnyLAN fué desarrollada originalmente por AT&T y HP y es administrada por el estándar de la IEEE 802.12. El estándar IEEE 802.12 es generalmente referido por su estándar de cableado, 100VG-AnyLAN.

Un punto clave de 100VG-AnyLAN es que esta no soporta CSMA/CD. Pero por otro lado si soporta un método de acceso llamado demand priority (prioridad en demanda) en la cual el HUB administra cuando y como el sistema puede accesar la red. Este soporta diferente niveles de prioridad, así puede garantizar que tareas con prioridad de tiempo reciban el acceso que ellas necesitan para un optimo rendimiento.

Este esquema permite poco o mejor dicho ningún espacio de contención ya que es el HUB quien determina cual estación de trabajo recibe acceso a la red. Cuando multiple transmisiones llegan al HUB simultaneas, la transmisión con la prioridad más alta es servida primero. Si la peticiones de las dos estaciones tienen el mismo nivel de prioridad, entonces se sirven ambas en corridas alternativas.

El método de acceso de prioridad en demanda usado por 100VG-AnyLAN puede ser ventajoso para algunos clientes porque este proveé una manera de garantizar prioridad para cierto tipo de tráfico en la LAN. Además, transmisiones enviadas a través del HUB no son retransmitidas (broadcast) a otras estaciones, así eliminando cualquier posibilidad de que alguien este escuchando (sniffing) nuestras transmisiones.

100VG-AnyLAN puede usar una gran variedad de cables para su estándar de alambrado (de aquí proviene el término AnyLAN). Por ejemplo, este puede usar las Categorías de UTP 3, 4, o 5 UTP, STP, o cables de fibra óptica.

**Apple LocalTalk**

LocalTalk es un tipo de red usado por Apple. Aunque no es un estándar IEEE, LocalTalk es importante ya que este usa un método de acceso a la red llamado Carrier Sense Multiple Access/Collision Avoidance (CSMA/GA). CSMA/CA específica que cada nodo debe informarle a los otros nodos de su intensiones de transmitir. Una vez los otros nodos han sido notificados, la información entonces es transmitida. Este acuerdo previene colisiones porque todos los nodos están al tanto de las transmisiones antes de que estas ocurran.

Ahora para resumir los métodos de acceso, IEEE 802.3 y Ethernet usan CSMA/CD, redes LocalTalk usan CSMA/GA, redes Token Ring usan un método basado en token, y las redes 100VG-AnyLAN usan prioridad en demanda (demand priority).

**Fiber Distributed Data Interfase (FDDI)**

FDDI es un estándar de LAN de alta velocidad. Fué desarrollado por el comité acreditador de Estándares X3T9.5 American National Standards Institute (ANSI). Al igual que el estándar IEEE 802.5 Token Ring, FDDI es basado en token.

Este estándar específica la subcapa de la MAC de la capa de Enlace de Data así como la capa Física para una LAN de 100-Mbps contra el reloj (sentido de como viaja la data), Token Ring, fibra óptica.

FDDI usa dos anillos que rotan la información para así proveer redundancia y permitir la red seguir funcionando si falla uno de los anillos. FDDI funciona bien sobre distancias sobre los 200 km (con un sólo anillo) con hasta 1,000 estaciones conectadas. En topologías de anillos dual, cada anillo se limita a solo 100 km. Como son usada para cubrir áreas metropolitanas o áreas específicas, una red FDDI puede ser clasificada

| Velocidad WAN | Medio | Protocolo(s) | Rango |
|---|---|---|---|
| 155Mbps+ | Fibra Optica | ATM | ˜10 a 100 km |
| 1.5 a 45 Mbps+ | Fibra Optica | Leased T/E 1-3 | ˜10 a 100 km |
| 64 Kbps a 2+ Mbps | Fibra Optica | X.25 | ˜10 a 100 km |
| 64 Kbps a 2 Mbps | Twisted pair | ISDN-30 (PRI) | Telefonica |
| 64 Kbps a 2+ Mbps | Twisted pair | Leased Frame Relay | Telefonica |
| 64 a 128 Kbps | Twisted pair | ISDN-2 (BRI) | Telefonica |
| 4 a 56 Kbps | Twisted pair | Modem | Telefonica |
| 1 a 4* bps | Aire | Semaphone | ˜1 km* |

como una MAN, Red de Area Metropolitana (Metropolitan Area Network).

Redes FDDI soportan ambos tipos de tráfico sincrónico y el asincrónico, y lo discutiremos más adelante.

## Estándares WAN

En estos últimos 20 años, la tecnología WAN ha progresado causa de que las compañías de telecomunicaciones han invertido en aumentar sus ancho de banda. Estas tecnologías incluyen el X.25, fast packet switching, Frame Relay, y ATM. Puede ser que usted ya este familiarizado con PPP, SLIP, y ISDN, los cuales también son métodos WAN.

* Son estimados, basados en la transmisión de data binaria, condiciones climáticas adecuadas.

Recuerde que el semaphore sufre de de perdidas de bits catastróficas después de la caída del sol. La distancia son aproximaciones ya que estas dependen de los materiales usados y las condiciones en las cuales ellas operan (Ej., niveles de interferencias). Para extender el rango de estas tecnologías, se usan repetidores de señales.

Una conexión de dial-up basada en modem difiere a las soluciones de conexiones, y mientras que el equipo necesario es substancialmente más económico y fácil de instalar, el intercambio es perdida de velocidad de acceso. La gran mayoría de los modems en la actualidad tansmiten desde los 14,400, 28,800 y 56,000 bits por segundos.

### Serial Line Internet Protocol (Protocolo de Internet de Línea Serial)

SLIP es uno de los primeros protocolos que fué desarrollado para enviar IP sobre líneas seriales como son los modems. SLIP es un simple conjunto de reglas que describen como los datagramas de IP pueden ser tramado para su transmisión sobre una línea serial.

SLIP fué la primera técnica desarrollada para implementar TCP/IP sobre línea serial pero nunca se formalizó como un estándar. La descripción definitiva puede se encuentra en el RFC 1055, “A Nonstandard for the Transmission of IP Datagrams over Serial Lines: SLIP.”

Algunas características de SLIP incluyen:
• Dependiente de MTU en requerimientos de enlace, Ej., tiempo de respuesta
• SD/ED caracter de escape (0xC0) (0xDB) si se encuentra presente en la data
• No revisa por error, reconocimiento (acknowledgment)
• Enlace asíncrono (debidamente, lento) solamente
• END mapeado a DB DC; ESC mapeado a DB DD

Básicamente, SLIP define como el inicio (start) y fin (end) de un datagrama de IP puede ser detectado por software leyendo bytes desde una conexión serial. Las reglas para encapsular datagramas de IP son simple y basadas en el uso de sólo dos caracteres de control:

- El caracter END (un byte único con el valor de 0xC0, o decimal 192) es usado para marcar el fin del datagrama. En la mayoría de los casos, la implementación SLIP usará este caracter para marcar el caracter de inicio (start) del datagrama aunque no es requerido.

- Para esconder el caracter de control usado por SLIP cuando ellos ocurren dentro del datagrama, el caracter de ESC (un byte único con valor de 0xDB, decimal 219) es usado. El caracter de END es transformado en una secuencia de dos bytes DB DC, y el caracter mismo de ESC es transformado en una secuencia de 2-byte DB DD.

El uso de SLIP ya no es lo que era antes, ha sido reemplazado en casi todas las plazas por el protocolo PPP.

**Limitaciones de SLIP**

Las limitaciones de SLIPs provienen principalmente de su simplicidad. La técnica básica de crear las tramas no permite para que más de un tipo de trama sea transmitida a la vez. Ethernet, por ejemplo, incluye un campo de identificación en el header de la trama. Esto significa que SLIP no puede compartir el enlace serial con más de una capa del protocolo de Red.

En un ambiente dial-up, es muy a menudo deseado no te tener un mapeo fijo de las direcciones IP al sistema actual. Pero, el protocolo SLIP no soporta ninguna idea de asignación de direcciones dinámicas; ambas puntas del enlace deben saber las direcciones IP entre si antes y durante la conexión.

No existe una contingencia para la detección ni corrección de errores, así que SLIP no funciona bien sobre líneas seriales ruidosas. Se asume que los protocolos del más alto nivel ejecutarán cualquier procesamiento de corrección de error necesario.

La variación de SLIP conocida como SLIP Comprimido (CSLIP) que incorpora la compresión Van Jacobson, el cual reconoce el hecho que la mayoría de header del IP es duplicado en cada paquete en todas las transmisiones. La compresión permite que los headers sean codificados en una punta del enlace y decodificada en el otro y así Típicamente lograr que sólo se envíen 8 bytes en vez de enviar 40.

**Protocolo de Punto-a-Punto (Point-to-Point Protocol, PPP)**

El PPP es una implementación más moderna que SLIP. Esta fué creada para sobrepasar muchas limitaciones del protocolo SLIP, y sus características pueden ser vistas como un superconjunto de las características de SLIP. Diferente de SLIP, es un estándar definido en el RFC 1548. Algunas de las características más importante de PPP son:
• Revisión Error
• Sincrónica/ Asincrónica
• Proveé autentificación débil (PAP) y fuerte (CHAP)

- Tres componentes por conexión:
  - Mecanismo de Encapsulación
  - Link Control Protocol (Protocolo de Control de Enlace, LCP)
  - Network Control Protocol (Protocolo de Control de Red, NCP)

PPP es un estándar basado en el derivada de la telefónica HDLC. El Protocolo LCP es usado para establecer, configurar, y probar la conexión data-link. Entonces el NCP es usado para negociar el Protocolo de la capa de red que será ejecutado sobre el PPP.

Por ejemplo, mientras se establece el IP, la dirección IP para cada host del enlace puede ser negociada. Hay mucho más además de esto para lo que es la especificación del PPP- variado esquemas de password/contraseñas, es sólo un ejemplo. Como PPP puede ser ejecutado virtualmente a cualquier velocidad que se requiera (enlace sincrónicos), esta es una escojencia muy generalizada para ambas conexiones punto-a-punto, la línea de modems telefónicas simple o las más rápidas enlaces de Internet (Ej., ISDN).

El PPP está en realidad compuesto por tres componentes:

**• Un mecanismo de encapsulación para los datagramas que son transmitidos**

Este mecanismo esta basado en el protocolo HDLC pero incluye un campo adicional que identifica el protocolo que está usando el enlace. Esto permite que más de un protocolo de red comparte este enlace.

**• El Protocolo de Control de Enlace (Link Control Protocol)**

Este protocolo se encarga de la configuración inicial (setup) y la operación del enlace a través de la negociación de varias opciones que toman lugar durante el establecimiento del enlace.

**• El Protocolo de Control de Red (Network Control Protocol)**

Este protocolo se concierne con la operación de configurar el protocolo de red y varia dependiendo del protocolo que este en uso.

La encapsulación de PPP tiene los siguientes campos:

| | |
|---|---|
| Flag | Byte 0x7E, usada al principio y fin de la trama |
| Addr | Byte 0xFF, siempre igual |
| Ctrl | Byte 0x03, siempre igual |
| Protocolo | Varia dependiendo del contenido, valores típico: 0x0021 datagramas IP, 0xC021 LCP y 0x8021 NCP. |
| CRC | Checksum |

## Configuración de Enlace en PPP

El LCP administra el estado de los enlaces:

**• Unidad de Recepción Máxima/Maximum Receive Unit**

Equivalente al MTU para las interfases tradicionales

**• Mapeo de Caracteres de Control (Control-character mapping)**

Este específica cuales caracteres de control ASCII deben ser mapeados a secuencias de escape.

**• Protocolo de Autentificación/Authentication Protocol**

Si existe cualquier protocolo de autentificación, este permite que los enlaces sean revisados mientras son establecidos, generalmente basado en alguna forma de mecanismo de contraseñas.

**• La Compresión de Cierto campos en la cabecera de la Trama/Compression of certain fields in the frame header**

Los enlaces PPP son altamente configurable. La configuración puede ser aplicada en dos etapas:

**• Durante el establecimiento del enlace, el LCP permite configuración por negociación entre los dos puntos de:**

- Unidad Máxima de Recepción
- Mapeo de Control de Caracter
- Protocolo de Autentificación
- Compresión de protocolo o dirección y campos de control

**• El NCP permite que aspectos de las características del protocolo de red del enlace a ser configurado. Para enlaces IP, es**

**controlado usando el Protocolo de Control del Protocolo de Internet (IPCP). Esto permite la configuración dinámica de:**

**• Dirección IP**

No es necesario saber la dirección IP de las dos puntas del enlace antes de establecerlo. Esto permite un uso mucho más flexible.

**• Protocolo de Compresión IP**

Muy a menudo usado para optimizar ejecución o rendimiento u es en particular importante en enlaces sobre seriales donde las velocidades de transmisión son más lentas que en las LANs típicas. El protocolo de compresión más usado comúnmente para TCP/IP es Van Jacobson Compression, el cual puede reducir los 40 bytes de los headers del IP y el TCP a niveles de 5 y 8 bytes.

**• WAN y otras tecnologías. Los pasos básicos para establecer PPP son:**

1.- Usted necesita un script para marcar el modem y luego ingresar su cuenta de usuario y contraseña.
2.- Pasar control al daemon del protocolo PPP.
3.- El daemon captura la dirección IP procedente del Proveedor de Servicios de Internet (ISP).
4.- Asociar puertos lógicos a puertos seriales.
5.- Establece a ifconfig.
6.- Establece el nombre de la interfase en la tabla de enrutamiento como la tura por defecto.

## Red Digital de Servicios Integrados/Integrated Services Digital Network (ISDN)

La tarea de un modem es tomar las señales digital que se originan desde el computador y convertir los flujos de data a análogas para transmitirla por las vía de intercambio telefónico, solo para ser reconvertidas a señales digital. ISDN funciona en esencia con el reemplazo de señales análogas a soluciones digital. La conexión es ya en esta etapa de punta a punta digital y puede soportar velocidades de transferencia de data que empiezan en los 64 Kbps. Debiéramos vernos en la necesidad de mayor velocidad, podemos usar múltiples conexiones ISDN en forma multiplexadas para obtener mayor velocidad.

Existen dos tipos de interfases usuarios ISDN, estas son Basic Rate interfase (BRI/ISDN2) y la de mayor ancho de banda Primary Rate interfase (PRI/ISDN30). Cada una de estas interfases le permite a la línea establecer un número diferente de canales. Existe también dos tipos de canales en específicos; estos son canales bearer (B), el cual escucha la data, y canales (D) que dan señales a la data, cual carga instrucciones y son acoplados con los canales B.

BRI proveé dos canales B de 64 Kbps y un canal de señal de 16 Kbps. PRI proveé 30 canales de 64 Kbps y uno de señal de 64 Kbps.

## Cable Modem (Telecable)

Los cable modems son una tecnología emergente para proveer acceso de alta velocidad al Internet con el uso de cables coaxial. Un sistema de televisión por cable por lo general tiene un 70 o más canales. Cada canal consume 27 Mbps de capacidad de descarga o downstream y 10 Mbps de capacidad de subir data o upstream. Todos los usuarios deben compartir el total del ancho de banda. El cable modem convierte señales entrantes a formato digital y entonces los conduce al computador a través del adaptador de Ethernet.

### *Cable Modems Simétricos y Asimétricos*

Hay dos maneras de usar un cable modem: simétrico y asimétrico. El modo simétrico permite que las velocidades de subida y de bajar datos sean iguales, este modo es usado muy rara vez. El modo asimétrico es mucho más común. En este caso, el canal de descarga es de más ancho de banda que el canal de subir. Las mayoría de las aplicaciones de Internet son asimétricas por naturaleza. Archivos de imágenes, sonido o video streaming tiende a ser de alto consumo en ancho de banda en la parte de descarga. Al contrario, las peticiones de URL y los mensajes de correo no son de alto consumo en ancho de banda en lo que respecta la subida de datos.

***Velocidades de Transferencia de Data***

El acceso a cable teóricamente permite velocidades de descarga de hasta 2 Mbps. Su promedio es de 450 KBps, mientras que las velocidades de subida es en promedio alrededor de 300 KBps, la cual refleja los principios fundamentales de esta tecnología. El cable modem no es de tecnología de punto-a-punto como son los modems estándar. Este es del tipo de tecnología bus, significando que todos los usuarios usen el mismo cable y tendrán que compartir su ancho de banda. A diferencia del acceso de un modem estándar, los cable modems permiten uso concurrente de las líneas alimentadoras. La velocidad depende de cuanta personas están conectadas a la vez de la misma vecindad que usan servicio de la misma compañía de servicio de cable. así pues que la transferencia de data no es de un valor constante.

Aunque la velocidad en promedio, que acabamos de mencionar, están muy alejadas de las máximas teóricamente posibles, ellas continúan aun siendo solo aproximadamente siete veces más rápidas que la velocidad de un modem estándar. Y aún más, redes de cable ya son un echo y existen, y por esto ya es relativamente fácil conectar un nuevo usuario a este servicio.

***Línea Digital de Subscriptor/Digital Subscriber Line (DSL)***

DSL es la alternativa a la tecnología de cable modem. DSL es una tecnología que trae información a alto ancho de banda al usuario final sobre líneas telefónicas de cobre regular. DSL es solo un miembro de la gran familia de los XDSL que incluye diferente variaciones de DSL, como son ADSL, HDSL, y VDSL. Bajo ciertas condiciones (Ej., que tan lejos se encuentra la conexión de la oficina telefónica que proveé el servicio), un usuario puede ser que tenga una velocidad de descarga de hasta 6 Mbps (donde el máximo en teoría es de 8.448 Mbps). El averaje de la velocidad de transferencia va desde 384 Kbps a 1.544 Mbps de descarga (desde el proveedor del servicio hasta el computador del usuario) y alrededor de 123 Kbps de subida (desde el computador del usuario al computador de ISP). Esto es mucho más mejor que las velocidades de un modem y hasta mejor que un cable modem en algunos de los casos. En averaje, cable modems ejecutan en el mismo rango que el DSL.

| Tipo de servicio | Proveedor-usuario (descarga de datos) | Usuario-proveedor (carga de datos) | Proveedor-usuario (descarga de datos) | Usuario-proveedor (carga de datos) |
|---|---|---|---|---|
| **ADSL** | 1.5 Mbps | 64 Kbps | 6 Mbps | 640 Kbps |
| **CDSL** | 1 Mbps | 128 Kbps | 1 Mbps | 128 Kbps |
| **RADSL** | 1.544 Mbps | 1.544 Mbps | 1.544 Mbps | 1.544 Mbps |
| **ISDL** | 128 Kbps | 128 Kbps | 128 Kbps | 128 Kbps |
| **RADSL** | 1.5 Mbps | 64 Kbps | 6 Mbps | 640 Kbps |
| **SHDSL** | No soporta | No soporta | 768 Kbps | 768 Kbps |
| **SDSL** | 1 Mbps | 1 Mbps | 2 Mbps | 2 Mbps |
| **VDSL** | 51 Mbps | 2.3 Mbps | 51 Mbps | 2.3 Mbps |

***¿Cómo esto Trabaja?***

Los modems tradicionales utilizan transmisiones análogas. así que, el monto de data máximo que se puede recibir usando un modem ordinario es 56 Kbps. El DSL, por lo contrario, es una tecnología que asume que data digital no necesita ser convertida a análoga y es transmitida directamente al computador en formato digital y permite a las compañías telefónicas usar un ancho superior sobre el mismo alambre de cobre. Parte del ancho de banda puede ser compartido y usado para transmitir señales análogas también. así que, podemos usar la misma línea para conectar tanto el teléfono como el computador para conectarnos.

***Tipos de DSL***

Como mencionamos anteriormente, existen variaciones del modelo DSL. Aquí mencionamos algunos de los tipos más comunes.

***ADSL***

La primera variación es la Línea de Subscriptor Asimétrica Digital (Asymmetric Digital Subscriber Line, ADSL), es una forma orientada de uso del hogar y los pequeños negocios, ADSL es llamado asimétrico porque su ancho de banda no es distribuido entre la subida y la descarga de la data. Solamente una pequeña porción del ancho de banda esta disponible en para la subida o los mensaje de la interacción del usuario. La mayoría del ancho de banda esta designado para la descarga de data como la mayoría de la Internet, y especialmente imágenes y multimedia, necesita mucho ancho de banda de baja. Recordemos que la mayoría de usuarios son particularmente sensibles acerca de su velocidades de baja.

***CDSL***

El Consumidor DSL (Consumer DSL, CDSL) es otra versión comercial del DSL. Es más lenta que el ADSL, pero el divisor (splitter) no tiene que ser instalado del lado del usuario. Esta tecnología es diseñada por la Corporación Rockwell y también esta dirigida a los usuarios del hogar de PC.

***HDSL***

Una de las primeras variaciones del DSL es el High bit-rate DSL (HDSL), la cual es usada para para transmisiones al nivel corporativo entre la compañía telefónica u el cliente. Esta es una tecnología completamente simétrica, lo cual significa que un monto igual de ancho de banda puede ser usado en ambas direcciones. Por esto es que la máxima descarga de data es menor que la de ADSL.

***IDSL***

ISDN DSL (IDSL) se encuentra más cercano al ISDN comparado por su velocidad de transferencia que el real DSL. IDSL nos solamente cerca de 128 Kbps de velocidad de bajar.

***VDSL***

VDSL es DSL de Muy alta velocidad (Very high bit-rate DSL) aún bajo desarrollo y ofrece velocidades de transferencia de data entre distancias relativamente cortas; por ejemplo, puede adquirir hasta 50 Mbps sobre las líneas de 1,000 pies de longitud. VDSL promete las velocidades más alta de todas las variantes del DSL.

***Factores que Determinan la Velocidad de Transferencia de Data***

El rango máximo de DSL sin repetidores (sin regenerar señal) es de 5.5 km, o 18,000 pies. A medida que la distancia entre el teléfono del usuario y el de la compañía telefónica se incrementa, la velocidad de transferencia de la data disminuye. Otro factor que influye la velocidad de transferencia es el tipo de cobre del alambre. Un ejemplo es, un cable de 24-gauge lleva la misma velocidad de data a una distancia mayor que un cable de alambre de gauge 26. Uno puede mantener la alta velocidad aunque se encuentren a más de 5.5 km.

Por lo general, el DSL es más confiable como el teléfono del usuario. A diferencia del cable modem, la velocidad del DSL permanece consistente a medida que más y más usuarios ingresan al sistema.

**X.25**

El X.25 fué desarrollado desde el protocolo 1822 del ARPANET, cual fué el esquema de intercambio de paquetes original del ARPANET. El X.25 se convirtió en estándar de ITU en el 1976. Es en la actualidad usado en transacciones por los cajeros automáticos, verificación de tarjetas de crédito, y muchas transacciones de otros puntos de ventas. El X.25 opera una velocidad de 56 Kbps o más lenta. Tecnologías más modernas de intercambio de paquetes (packet-switching) pueden hoy en día ejecutar mejor que el X.25, pero es aún muy usado por todo el mundo. El X25 asegura el transporte de data libre de errores a través de la revisión de errores en muchos puntos de la ruta tomada por la data.

***Rápido Intercambio de Paquetes/Fast Packet Switching***

Fast packet switching se refiere a dos tipos diferentes de trasmisión a través de redes del tipo maya de intercambio. Con fast packet switching, las tareas de corrección de error, secuencia de paquetes, y reconocimiento no son ejecutados por la red. Fast packet switching es la responsabilidad del sistema destino; y ya que la red tiene menos tareas que llevar acabo puede agilizar la transferencia de la data.

La eliminación de corrección de errores en las capas inferiores ampliamente aumenta los niveles de ejecución. La tecnología de fast packet switching es implementada al nivel de la subcapa de MAC en la capa de Enlace de Data, contrario al X.25, el cual es implementado en la capa de Red.

Ejemplos de la tecnología fast packet-switching are Frame Relay y ATM.

***Frame Relay***

El Frame Relay es de tecnología fast packet-switching que usa fibra óptica para el cableado digital. Este usa paquetes de longitud variadas (variable-length packets) y permite conexiones a más alta velocidad usando facilidades compartidas de la red. No ofrece soporte extenso de revisión de error y reconocimiento, como si hace el X.25. Como lo implica el nombre, Frame Relay es un servicio de relevo (relay), y el X.25 es un servicio de paquete.

Si su compañía desea conectarse a una red de Frame Relay, su compañía de servicios telefónicos debe conectarlo a un puerto Frame Relay, o Punto de Presencia (Point of Presence, POP). El POP debe ser un proveedor de servicios de Frame Relay. Un puerto de Frame Relay es creado para usted, así dandole acceso a la red Frame Relay.

Redes Frame Relay usan Circuitos Permanentes Virtuales (Permanent Virtual Circuits, PVCs). Estas conexiones lógicas, dedicados, de punto a punto son usadas para la transferencia de data. Una vez establecida una conexión PVC, este existe hasta que la transacción, transmisión o el servicio es terminado. así que, Frame Relay es una red basada en software ya que este comparte la red física con otras redes Frame Relay.

Frame Relay utiliza el ancho de banda en demanda, esto significa que los clientes de Frame Relay pueden elegir el monto de ancho de banda que ellos necesitan. Cada puerto Frame Relay tiene su propia velocidad de puerto, por lo general esta velocidad cae en el rango de 64 Kbps a 1.544 Mbps (T1). Frame Relay es implementado en la subcapa de la MAC de la capa de Enlace de Data.

***Modo de Transferencia asincrónico/Asynchronous Transfer Mode (ATM)***

ATM utiliza celdas de tamaño fijo (en vez de como las de tamaño variado de paquetes del Frame Relay) y PVC para soportar data así como video y voz. ATM puede ser usado por ambas redes tanto LAN y WAN, pero es más comúnmente usado como un Internet backbone. El protocolo ATM esta definido por el ITU (International Telecommunication Union) y el Foro especializado de ATM.

ATM es referido a una tecnología de relevo de celda ya que esta organiza data en celdas de longitud fija de 53-byte. Estas celdas son transmitida y segmentadas dependiendo del tipo de data que esta siendo enviada. Por ejemplo, el ancho de banda es asignado dependiendo de la aplicación en uso.

Como el ATM usa celdas de tamaño fijo, es más rápido que el Frame Relay. Dispositivos de switching no necesitan ubicar el principio o el fin de cada celda; estas celdas son todas de un mismo tamaño.

Aunque ATM Típicamente opera a velocidades que van desde los 155 Mbps hasta 622 Mbps, este tiene un potencial de velocidad de 1.2 Gbps. Las tecnologías FDDI y T3 están implementando elementos de ATM.

Pero, no todos los medios soportan las capacidades de ATM. Similar al Frame Relay, el ATM es implementado en la subcapa de la MAC de la capa de Enlace de Data.

***Sistema T-Carrier***

El sistema T-carrier es un formato de transmisión digital Norte Americano que proveé servicios dedicados y de líneas privadas para transmisiones de voz digital y de data a velocidades que llegan hasta los 45 Mbps. Los servicios T-carrier por lo general son usados para conectar una LAN a una WAN, como es la red de una compañía al Internet o a una red Frame Relay.

| T-Carrier | Data Transfer Rate |
|---|---|
| T1 | 1.544 Mbps |
| T2 | 6.312 Mbps |
| T3 | 45.736 Mbps |
| T4 | 274.176 Mbps |

***T1***

La T1 es un Servicio de alquiler de línea digital COMÚN que proveé un ancho de banda de 1.544 Mbps. Cada línea T1 soporta 24 canales de 64 Kbps. Cada uno de los 24 canales en un circuito T1 puede cargar la transmisión de voz o data.

| E-Carrier | Data Transfer Rate |
|---|---|
| E1 | 2.048 Mbps |
| E2 | 8.448 Mbps |
| E3 | 34.368 Mbps |
| E4 | 139.264 Mbps |
| E4 | 565.148 Mbps |

T1 Fraccional, mejor conocidas como FT1, permite que un cliente alquile canales independiente en vex de una línea t1 completa. Este servicio rinde velocidades de transferencia de data de 1.536 Mbps (un adicional de 64 Kbps es usado para la sobrecarga o overhead).

Para conectar la línea T1 a su red LAN, usted necesita los siguientes sistemas:

**CSU**
: Primer punto de contacto del cableado de la T1; este diagnostica y prepara la señal en la línea para la LAN.

**DSU**
: Se conecta al CSU y convierte las señales de la LAN a formato de señales de la T1.

**Multiplexer**
: Proveé un mecanismo para cargar multiple canales de voz y data en una sola línea.

**Router**
: Proveé la interfase entre la LAN la línea T1

***T2***

T2 es una especificación interna la cual es el equivalente de cuatro líneas T1. Líneas T2 proveen un ancho de banda de 6.3 Mbps. Línes T2 no son ofertadas al publico en general.

***T3***

Una T3 es equivalente a 28 circuitos T1 y proveen un ancho de banda total de de 44.736 Mbps. T3s fraccionadas permiten al cliente alquilar menos que la total velocidad de una T3 completa. La clasificación de Señal Digital (Digital Signal, DS) usada en Norte America, Japón, Korea, y otros paises proveen una estandarización para los niveles de señales digitales y son sinónimos con el sistema T-carrier. En la siguiente tabla mostramos las velocidades de transferencias del sistema T-carrier.

***Sistema E-Carrier***

El sistema E-carrier es un formato de transmisión digital Europeo similar al sistema T-carrier Norteamericano. Cada velocidad de transmisión es un múltiplo del formato E1, el cual opera a 2.048 Mbps. En esta siguiente tabla le mostramos el lista de las velocidades de este carrier.

## DIRECCIONES DE INTERNET

Como mencionamos anteriormente, para poder pasar mensajes de un computador a otro, cada computador deberá tener su propia dirección de red. Ya hemos visto la dirección MAC, la cual es usada por la capa de Enlace de Data. IP tiene su propio mecanismo de asignación de direcciones. Debido a la separa-

ción de las capas de los protocolos, la capa de Internet no puede por lo general usar ni depender de la capa de Enlace de Data para que le proveé de sus direcciones. Además, el esquema de asignación de IP divide las direcciones en un número de red y un número de nodo, mientras que la dirección MAC no puede.

Cubriremos el formato de las direcciones del Protocolo de Internet, como ellas están estructuradas, y como las direcciones IP son elegidas y obtenidas. También discutiremos las intranets. Los siguientes tópicos son discutidos en esta sección:

- Capa de Internet
- Header (Cabezal) de IP
- Direcciones IP
- Clases de Direcciones de Internet
- Reglas de Direcciones IP
- Eligiendo las Direcciones IP
- Intranets
- Subredes

**Capa de Internet**

La capa de Internet proveé el medio para la comunicación entre los computadores en una red. Para habilitar esto, la capa de Internet ejecuta la función de enrutar paquetes de una red a otra hasta que estos lleguen a su destino. Para poder efectuar sus funciones de enrutar, la capa de Internet puede asignar direcciones a los hosts en una red y a las redes mismas.

La capa de Internet proveé servicios a la capa más alta de Transporte. Los servicios que proveé están se conciernen por lo general con la transferencia de paquetes entre los hosts.

La capa de Red del modelo OSI cae por debajo de la capa de Enlace y encima de la capa de Transporte. La capa de Enlace de Data puede ser vista como si tuviese dos capas propias. Una de estas, la MAC (Media Access Control), maneja las direcciones MAC de 6-byte que discutimos anteriormente, contiene los manejadores o drivers para las tarjetas adaptadoras de red, y maneja la lógica de colisiones. La otra capa, la capa Control Lógico de Enlace (Logical Link Control, LLC), yace más arriba de ella, establece y termina los enlaces, y establece las direcciones usadas por las varias capas del modelo OSI para comunicarse entre ellas.

La capa de Red proveé servicios a la capa de Transporte (esta comunica sus datagramas), y esta usa los servicios de la capa de Enlace de Data (al transferir paquetes en la red local).

**IP Header**

La versión actual del IP, la versión cuatro (IPv4), tiene un header que consiste a menudo campos fijos en el header, dos direcciones, y opciones. La longitud del header de un paquete IPv4 es generalmente de 160 bits (20 bytes) al menos que existan opciones. La última versión del IP, la versión seis (IPv6), la discutiremos brevemente.

Aquí le presentamos una ilustración del header del paquete IPv4. El header consiste de seis WORDS (termino informático de almacenaje parecido al CHAR) de 32-bit, la cual es 24 bytes (con opciones presente). Si la data no llena un WORD de 32-bit, particionamiento de bit es a menudo usado para completarlo.

El header del paquete IP contiene varios campos muy importantes:

**Versión (4 bits)**

Este campo identifica la versión IP, actualmente es la versión 4.

**Header Length (4 bits)**

Este específica la longitud del header del paquete IP. Valores de la longitud del header son expresados en el número de WORDS de 32-bit en el header, el cual es por lo general cinco al menos que existen opciones presentes.

**Service (8 bits)**
Este campo indica la confiabilidad, retraso de presidencia, y el resto de los parámetros. También es conocido como el campo del Tipo de Servicio (Type of Service, TOS).

**Datagram Length (16 bits)**
Este campo define la longitud total del paquete, incluyendo el header, en bytes. La longitud del datagrama no incluye el header usado en la capa de Acceso a la Red (Ej., header del Ethernet).

**Datagram ID Number (16 bits)**
Esta identificación única de un paquete para los propósitos de fragmentación y reensamblaje. Este número único es copiado en cada fragmento de un datagrama en particular para que así este pueda ser ensamblado.

**Flags (3 bits)**
Este campo es sólo usado para fragmentación y reemsamblaje.

**Fragment Offset (13 bits)**
Este indica donde en el paquete este fragmento cae.

**Time To Live (8 bits)**
Este es medido en intervalos de un segundo, con un máximo de 255. Este campo es conocido como el campo TTL. Los routers por lo normal eliminan un segundo del campo TTL, en vez de calcular los segundos reales, así que el campo TTL es a veces conocido como el campo de salto o HOP Field.

**Protocol (8 bits)**
Este campo define el próximo nivel de protocolo que es el que recibirá el campo de data en el punto de destino. Si el campo de protocolo es marcado 1, este es un paquete ICMP ; si es 6, este es TCP; si es 17, este es UDP. Fijese que aunque el ICMP y el IGMP son incorporados en la capa de Internet, ellos son encapsulados en paquetes de IP.

**Header Checksum (16 bits)**
Este campo es usado para error de detección. El checksum calcula solamente el header del IP.

**Source Address (32 bits)**
Este campo calcula la dirección IP del sistema de origen.

**Destination Address (32 bits)**
Esta identifica la dirección IP del sistema de destino o final.

**Opciones**
Este campo indica información del paquete, como son:
- Seguridad/Security
- Enrutamiento de fuente Estricto o Ligero/Loose o strict source routing
- Reportaje de error/Error reporting
- Estampar/Timestamping
- Persecución de errores/Debugging

Por ejemplo, la opción de enrutamiento le permite al que envía especificar la ruta que un paquete a de seguir a través del Internet. Ambas la estricta y la permisivas especifican la vía de enrutamiento.

Enrutamiento permisivo de origen permite que múltiples saltos de red entre direcciones de Internet sucesivos en la lista. El enrutamiento estricto implica que la dirección de Internet especifique la ruta exacta que un paquete debe seguir para llegar a su host de destino; resulta un error si un route no puede reenviar o forward el paquete al nodo especificado.

## Direcciones IP

La clave de las capacidades de internetworking del protocolo de Internet esta en su estructura de su direcciones:

- Cada nodo en un internetwork de TCP/IP debe tener una dirección única. Como esta dirección es interpretada por IP, es conocida como la dirección IP.
- Las direcciones incluyen un componente que identifica la red actual a la que el host está conectado además de una porción que identifica el host individual.
- Este concepto puede ser comparado con el de la dirección de una casa, donde parte del componente de la dirección identifica el nombre de la calle (la red) y la otra parte el número de la casa (el host).

Para asegurar que cada usuario en el Internet tenga una dirección IP única, una autoridad central llamada

el ICANN (Internet Corporation of Assigned Names and Numbers), anteriormente conocida como IANA (Internet Assigned Numbers Authority), asigna todos las direcciones de Internet. IANA era financiada y supervisada por el gobierno de los Estados Unidos. La ICANN es una organización sin fines de lucro que opera a nivel internacional, responsable de asignar espacio de direcciones numéricas de protocolo de Internet (IP), identificadores de protocolo y de las funciones de gestión o administración del sistema de nombres de dominio de primer nivel genéricos (gTLD) y de códigos de países (ccTLD), así como de la administración del sistema de servidores raíz. Aunque en un principio estos servicios los desempeñaba Internet Assigned Numbers Authority (IANA) y otras entidades bajo contrato con el gobierno de EE.UU., actualmente son responsabilidad de ICANN.

Como asociación privada-pública, ICANN está dedicada a preservar la estabilidad operacional de Internet, promover la competencia, lograr una amplia representación de las comunidades mundiales de Internet y desarrollar las normativas adecuadas a su misión por medio de procesos "de abajo hacia arriba" basados en el consenso.

La mayoría de las direcciones de Internet contiene una parte que es la red y la otra parte que es el host. La parte que es la red precede la parte que representa el host:

**porción red.porción host**

Las direcciones de Internet son especificadas por cuatro campos, También llamados octetos, separados por puntos:

**campo1.campo2.campo3.campo4 o (xxxx.xxxx.xxxx.xxxx)**

| Valor del BIT | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|

Ellos son típicamente escritos en una anotación de punto decimal. Cada campo tiene un valor de 0-255 como demostramos en el siguiente ejemplo de una dirección de Internet:

**200.42.200.136**

| 10101001 | 10100111 | 11001010 | 00000010 |
|---|---|---|---|
| 169. | 167. | 202. | 2 |

En este ejemplo, la parte que representa la red es 200.42.136, y la porción que es el host es el 136. Para ayudar a distinguir entre la parte de la dirección que es red y la parte que es cliente, las direcciones de Internet están divididas en clases, las cuales describiremos en este mismo capítulo más adelante.

**Formato Binario vs. Decimal**

Las direcciones son llamadas de 32-bit porque cada campo es realmente un byte, y un byte es igual a 8 bits. Como una dirección IP tiene 4 bytes, el total es de 32 bits.

**8 + 8 + 8 + 8 = 32**

Para poder determinar el valor en bit de una dirección de Internet, la dirección deberá ser convertida desde su formato decimal a si formato binario. Formato binario es la combinación de ceros y unos que las computadoras usan para procesar información. El equivalente binario es determinado calculando el valor de cada bit dentro de cada byte, de izquierda a derecha.

Si el valor binario de una dirección IP es 01111001, usted puede determinar el valor decimal simplemente sumando el valor correspondiente a los bits con valor de 1. Por ejemplo:

**01111001 =0+64=32+16+8+0+0+1=121**

Tu puedes hacer esto para cada byte en direcciones de Internet de 32-bit. Por ejemplo:
**10000011 1110001000001000 11001000 = 131.226.8.200**

**Ejercicios 2-3: Convertir direcciones de Internet en valores decimales y binarios.**

En este ejercicio, usted deberá convertir direcciones de Internet de binarias a decimales y vice versa. Escriba su respuesta en el espacio en blanco. Las respuestas para este ejercicio están en el Apéndice B.

A.- Convertir de valores de binarios a decimal

1. 01111000000000000100000011 11110000

2. 11000011 01010101 10011001 11010

B.- Convertir de valores de decimales a binarios

3. 207.199.32.205

| Clase A: Rango de 0.0.0.0 al 127.255.255.255 | | | |
|---|---|---|---|
| Valor Binario Inicial | 0 | Red (1 Byte) | Host (3 Bytes) |
| 126 Redes | | | 16,777,214 Hosts |

| Clase B: Rango de 128.0.0.0 al 191.255.255.255 | | | | |
|---|---|---|---|---|
| Valor Binario Inicial | 1 | 0 | Red (2 Byte) | Host (2 Bytes) |
| 16,384 Redes | | | | 65,534 Hosts |

| Clase C: Rango de 192.0.0.0 al 223.255.255.255 | | | | | |
|---|---|---|---|---|---|
| Valor Binario Inicial | 1 | 1 | 0 | Red (3 Byte) | Host (1 Bytes) |
| 2,097,152 Redes | | | | | 254 Hosts |

| Clase D: Rango de 224.0.0.0 al 239.255.255.255 | | | | | |
|---|---|---|---|---|---|
| Valor Binario Inicial | 1 | 1 | 1 | 0 | Red Multicast (4 Bytes) |

| Clase E: Rango de 240.0.0.0 al 247.255.255.255 | | | | | | |
|---|---|---|---|---|---|---|
| Valor Binario Inicial | 1 | 1 | 1 | 1 | 0 | Experimental/Reservada para uso en el Futuro |

3. 151.2.254.60

**Clases de Direcciones de Internet**

Sin un sistema de clasificación, las 3,720,314,628 direcciones posibles de Internet pueden no tener estructura. Para proveer una estructura, los diseñadores de los protocolos de Internet definieron cinco clasificaciones de direcciones de Internet, de las cuales tres son conocidas y usadas comúnmente:

- Grandes redes, con muchos cientos de miles de hosts, conocidas como redes de clase A.
- Instalaciones de tamaño medio, conocidas como redes de clase B.

• Redes pequeñas con unos cuantos hosts, conocidas como redes de clase C.

Las clases cuatro y cinco están reservadas para usos especiales:
• Multicast, conocidas como clase D.
• Uso futuro y de prueba, conocidas como clase E.

Las cinco clases de redes usan la forma básica de direcciones de Internet. Las clases pueden ser determinadas por la observación a el primer byte de una dirección de Internet.

Las características de cada clase son detalladas a continuación, seguidas por una explicación de cada una:

Antes de leer acerca de las clases de direcciones, por favor note que ni la porción de red ni la porción de hosts de una dirección IP pueda contener todo ceros y ni todo unos en formato binarios. En valores decimales, 255 usualmente significa broadcast, y un valor 0 identifica la red. Estudiaremos más acerca del rol de las direcciones IP más adelante en este capítulo.

**Direcciones de Clase A**

Las direcciones de clase A usan los primeros 8 bits para la porción de la red y el sobrante de 24 bits para la porción de hosts. Estas proveen el potencial para 126 redes con 16,777,214 hosts. El primer byte específica el número y clase de la red, este puede clasificar de 1 a 126 (127 esta reservado para direcciones loopback). El primer bit de una dirección de una red clase A es siempre un bit 0.

Aquí le presentamos un ejemplo de una dirección de clase A (el primer byte es la dirección de la red):

**121.1.1.32**

El bit equivalente es:

**01111001 00000001 00000001 00100000**

**Direcciones de Clase B**

Las direcciones de clase B usan 16 bits para la porción de red y 16 bits para la porción de hosts. Ellas proveen el potencial para 16,384 redes y hasta 65,534 hosts por cada una. Los primeros 2 bytes especifican el número de la red y la clase, el primer byte puede variar de 128 a 191. Los primeros 2 bits de una red clase B son siempre 10.

Lo siguiente es un ejemplo de una dirección clase B (los primeros 2 bytes son las direcciones de la red):
**168.100.1.32**

Esto es equivalente a:
**10101000011001000000001 00100000**

**Direcciones de Clase C**

Las direcciones de clase C usan 24 bits para la parte de la red y 8 bits para la porción de hosts. Ellas proveen el potencial para 2,097,152 redes con hasta 254 hosts por cada una. Los primeros 3 bytes especifican el número y la clase de la red, el primer byte puede variar de 192 a 223. Los primeros 3 bits de una red clase C siempre son 110.

Lo siguiente es un ejemplo de una dirección clase C(los primeros 3 bytes son la dirección de la red):
**205.96.224.32**

El equivalente en bit es:

**11001101 01100000 1110000 000100000**

**Direcciones de Clase D**

Las direcciones de clase D soportan multicasting. Con multicasting, un datagrama es dirigido para un grupo que esta identificado por una dirección de red solamente (sin la existencia de porciones de hosts). El primer byte puede variar de 224 a 239. los primeros 4 bits de una red clase D siempre son 1110.

Lo siguiente es un ejemplo de una dirección clase D (los cuatro bytes son la dirección de la red):

**230.5.124.62**

El equivalente en bit es:

**11100110000001010111110000111110**

**Direcciones de Clase E**

Las direcciones de clase E están reservadas para usos futuro. El primer byte puede variar de 240 a 247. Los primeros 5 bits de una dirección de red clase E siempre son 11110.

Podemos ver que pueden haber relativamente pocas y distintas direcciones de Clase A pero muchas direcciones de clase C. En practica, esto es un reflejo exacto de los tipos de redes que se encuentran en el Internet, muy pocas grandes organizaciones y muchas más de las pequeñas.

Es posible de seguir dividiendo una simple red en un número de pequeñas subredes lógicas con el uso de la porción host de la dirección como una dirección de subred. Este topico es cubierto en más detalles en el transcurso del capítulo.

La escasez de números de direcciones IP disponibles para asignar a usuarios nuevos es en la actualidad un tema de discusión y uno de los problemas mayores con IPv4. IPv6 esta diseñado para solucionar estas inconvenientes.

**Reglas del Direccionamiento IP**

Las direcciones de Internet tienen que seguir fuertes alineamientos para fusionar apropiadamente. Aunque tenemos cubiertos los rangos de direcciones de las clases A, B y C, no todas las direcciones con estos rangos pueden ser usadas como nodo de direcciones de red. Esta sección cubrimos las excepciones.

Todas las reglas del direccionamiento IP están basadas en las reglas básicas de que ni la porción de red o la de host pueden ser todos binarios, unos o ceros.

**Direcciones Broadcast**

Las direcciones Broadcast son usadas para enviar mensajes a todos los nodos de la red. Las porciones de direcciones IP de red y/o hosts todas son binarias, las cuales generalmente coinciden con el valor decimal 255. Ellas son usadas solo para direcciones de destinación y no pueden ser usadas para direcciones de origen. Existen los siguientes cuatro tipos:

- **Broadcast limitado: 255.255.255.255**

  Ambas, la porción de red y host consisten de unos binarios. Este tipo es usado para configurar hosts cuando estos se inician/bootean. Por ejemplo, una computadora sin una dirección IP puede broadcast para obtener una dirección (por ej., desde un servidor DHCP o BootP).

- **Broadcast dirigido por la Red; netid.255.255.255**

  Este es usado para broadcast a todos los hosts en una red. Por ejemplo, si la cantidad de direcciones IP de tu red es 192.34.200 y la cantidad de hosts es 12, tu computadora puede broadcast mensajes a todos los hosts de la red usando la dirección de destino 192.34.200.255.

- **Broadcast dirigido por la Subred**

  Si una red esta dividida en muchas subredes, un broadcast puede ser limitado a los hosts dentro de una subred. Las subre-

des serán discutidas más adelante en este mismo capítulo.

• **Broadcast dirigido para todas Subredes**

Si una red esta dividida en varias subredes, un broadcast puede ser enviado a todos los hosts dentro de todas las subredes en la red. Este tipo de broadcast ya es obsoleto, multicasting (ve direcciones de clase D) es preferido.

## Direcciones de Red

Las direcciones de red son usadas por los routers para identificar una red. La porción red consiste en la dirección de la red, pero la porción de los hosts consiste de ceros binarios (netid.0.0.0). Por ejemplo, la dirección de red 192.168.3.0 no puede ser usada como una dirección de host.

***Caso especial de Direcciones de Origen***

El caso especial de direcciones de origen es usado cuando una computadora no tiene una dirección IP. Este es usado solo durante el proceso de inicialización. En un caso especial de direcciones de origen, las porciones de red y los hosts de las direcciones IP son todas ceros binarios (0.0.0.0). Esta dirección es usada cuando una computadora inicializa y pide una dirección IP (por ej: de un servidor DHCP o BootP). Aunque la computadora broadcast una petición para la asignación de una dirección IP, esta dirección de origen es inicializada 0.0.0.0 hasta que se le asigne una dirección IP.

***Direcciones Loopback***

La dirección loopback 127, no puede ser usada como una dirección de red. Esta dirección permite a un cliente y servidor en el mismo host comunicarse entre si. La dirección loopback es ideal para examinar y diagnosticar. Por ejemplo, si tu computadora es un servidor Web y tu entras “http://127.0.0.1” en un navegador Web (en este caso el cliente), tu puedes accesar al sitio Web (aunque el servidor esta en el mismo sistema). La dirección loopback también puede ser usada para examinar la funcionalidad del TCP/IP con el utilitario ping.

Para sistemas UNiX y NT, la dirección loopback esta listada en el archivo /etc/hosts y es Típicamente 127.0.0.1 con el nombre “localhost” asignado.

## Ejercicio 2-4: Determinar las Clases y Direcciones IP Validas

En este ejercicio, usted deberá determinar la clase de cada dirección IP y además que sea una dirección IP válida para un computador. Si no lo es, explicar porque. Las soluciones para este ejercicio están localizadas en el Apéndice B.

| | Dirección IP | Clase | Valida? Si o No | Si no es valida, ¿por qué no? |
|---|---|---|---|---|
| 1. | **192.23.111.8** | | | |
| 2. | **10.1.1.256** | | | |
| 3 . | **148.108.62.95** | | | |
| 4. | **127.0.0.1** | | | |
| 5. | **245.255.123.49** | | | |
| 6. | **100.54.100.90** | | | |
| 7. | **162.34.0.0** | | | |
| 8. | **127.65.18.191** | | | |
| 9. | **1.1.1.1** | | | |
| 10. | **208.152.84.255** | | | |
| 11. | **225.37.257.34** | | | |
| 12. | **255.255.255.255** | | | |

## Direcciones IP Reservadas

El ICANN tiene reservado un espacio de tres bloques de direcciones IP para redes privadas (como esta definida en RFC 1918):

- 10.0.0.0 por medio de 10.255.255.255
- 172.16.0.0 por medio de 172.31.255.255

• 192.168.0.0 por medio de 192.168.255.255

El ICANN sugiere que compañías usen estos IDs de redes si la compañía encaja dentro de una de las siguientes categorías:

- Sus hosts no requieren acceso a otras empresas o hosts de Internet.
- Las necesidades de Internet de los hosts pueden ser manejadas por gateways (ej: gateways de la capa de Aplicación). Por ejemplo, los hosts solo requieren servicios de Internet limitados, tales como correo electrónico, FTP, grupos de noticias y exploradores Web.

Estas direcciones de redes privadas no tienen un significado global. Por lo tanto, se espera que los routers de Internet rechazen (filtren) información acerca de ellos (el rechazo no puede ser tratado como un error de protocolo de enrutamiento).

Los beneficios de usar direcciones de redes privadas incluyen:

- Conservación de direcciones IP globalmente únicas cuando no se requirie.
- Más flexibilidad en el diseño de soluciones empresariales a causa de grandes espacios de direcciones.
- Prevención de choque de direcciones IP cuando una empresa obtiene conectividad a Internet sin recibir las direcciones desde ICANN.

Las desventajas de usar direcciones de redes privadas incluyen:

- Posible reducción de la flexibilidad para accesar a Internet. Si la compañía eventualmente decide proveer conexión a Internet para algunos o todos los hosts, se deberá renumerar parte o toda la compañía.
- Si la compañía absor con otra compañía y todos los hosts usan direcciones de red privadas, probablemente se necesitara combinar varias redes privadas dentro de una. Las direcciones dentro de la red privada combinada pueden no ser únicas y tendrá que renumerar los hosts para acomodar direcciones IP idénticas.

**Eligiendo Direcciones IP**

Cuando la red de un edificio usa TCP/IP, a cada nodo se le debe asignar una dirección. Si la red nunca esta conectada a una gran internetwork, entonces la red puede usar uno de los rangos reservados para uso interno privado.

Si la red es conectada a una gran internetwork, igual que el Internet, los componentes de una dirección de red local deben ser obtenidos desde una autoridad central administrativa. Esta autoridad puede establecerse iniciando routing para la nueva red. Para el Internet, las direcciones de redes de donde anteriormente obtuvieron directamente, desde el InterNIC en Stanford. Este no es el gran caso; dentro del orden para reducir el tamaño de las tablas de routing, en las sinClases InterDominantes de Enrutamiento (Classless InterDomain Routing (CIDR)) las provisiones tienen agrupadas direcciones de Internet por geografía. Ellas están cerradas para que la obtención de cualquier cosa que no sea un grupo de direcciones clase C sea imposible, CIDR hace efectivamente la noción de clases de direcciones redundantes.

Las direcciones IP son básicamente asignadas por tu ISP o tu conexión up-link. Solo los sitios backbone son atractivos para ser concedidas direcciones IP desde el InterNIC. Cada persona debe tomar su dirección IP desde su ISP. Esto se hace un pequeño problema, mientras cambias tu ISP te va a requerir que tu cambies tu rango de direcciones IP en la mayor parte de las circunstancias. Afortunadamente, DNS usualmente hace esto un punto de discusión, así los sitios son usualmente direccionados por nombre no por número.

Una vez que una dirección de red ha sido asignada, las direcciones de los hosts pueden ser colocados por el administrador de la red o un protocolo automatizado como DHCP.

**Intranets**

Una intranet es la red interna de cualquier organización. El termino "Intranet" fué inventado para referirse a redes internas que usan los mismos protocolos como el Internet, ej: TCP/IP, Este también implica que los basados en IP, los protocolos de alto nivel son usados también, como es el HTTP. Otros usos de los protocolos TCP/IP hay mucha diferencia entre una LAN y una intranet.

Las intranets son distintas, ellas casi siempre implican accesos restringidos para la información y los hosts. Muchas intranets no son conectadas a cualquier sistema de redes exterior, e históricamente esta es como todas las grandes organizaciones edificadas sobre redes de computadoras, usan cualquier tecnología de red que ellos prefieran.

En el mundo moderno de cualquier manera esto incrementa comúnmente para organizaciones que quieren conectar sus intranets a el Internet en una forma limitada, sea para proveer información de ayuda en la intranet a clientes distantes o para permitir a usuarios de su intranet beneficiarse de servicios en el Internet y para distribuir información entre sitios unidos.

Porque la conexión de cualquier red a otras empresas de seguridad unidas es usualmente hecha por la conexión indirecta del intranet al Internet, por ejemplo, usando un firewall. Aquí, el defiende la intranet del resto del Internet mientras permite ciertos tráficos relacionados a través de ciertos servicios.

Aveces organizaciones conectan directamente sus intranet al Internet y luego confían en controles de acceso de servicios personales para restringir quien puede usar sus servicios de intranet.

Las preguntas de todas estas empresas son: ¿Cuáles direcciones IP usar para la intranet? ¿Necesitamos el uso de direcciones que sean únicas en el Internet (con el costo que puedan contraer)?

**Direccionamiento de Intranet**

TCP/IP es usado en muchas redes que no están conectadas al Internet. Usualmente, donde estas nunca esperan ser conectadas, y una clase de dirección IP apropiada para la red que es seleccionada al azar. Con la explosión de ventajas en el Internet, muchas organizaciones son consideradas una conexión; pero porque direcciones IP deben ser globalmente únicas, ellos se encuentran con el problema de cambiar completamente el esquema de direccionamiento usado en la red.

Un servidor proxy puede ser usado para transmitir conexiones desde una red interna o externa, representando una apropiada forma de traducción de direcciones. Todo aquello que es requerido para una simple máquina. El único hosting, el servidor proxy, tiene una dirección de Internet reconocida en la interfaz externa.

El enrutamiento puede aun ser afectado donde la clase de direcciones usada internamente pertenezca a otra organización que ya este conectada a Internet. Por ejemplo, por accidentes, la dirección IP perteneciente a Micro-Soft puede estar siendo usada. Los enrutadores en la red interna pueden enrutar todos los paquetes destinados para esta clase de direcciones a máquinas locales mejor que a través de el servidor proxy, o el mismo servidor proxy puede enrutar los paquetes despalda a la red interna. RFC 1918 reconoce la practica del uso de direcciones privadas y la codificación del problema por configuración al lado de ciertas direcciones IP. (red clase A 10, redes clase B 172.16 a 172.31 y redes clase C 192.168.0 a 192.168.255). esto es útil porque estas direcciones nunca pueden ser usadas por redes en el Internet.

Aun hay algunas deficiencias en este esquema. Dos organizaciones o departamentos pueden tener usadas algunas clase de direcciones desde este esquema, luego ellos pueden necesitar absorber sus redes (antes de

sobretomar, tal vez); ellos pueden entonces necesitar configurar los hosts con conflictos.

**Ejemplo de Direccionamiento de Intranet**

En el siguiente ejemplo de direccionamiento de una intranet, la red de Clase A reservada 10.0.0.0 puede ser usada para una gran intranet corporativa. Esto permite un espacio amplio de direcciones para la red interna. De cualquier manera los usuarios querrán aun conectarse a Internet. Esto se logra colocando un gateway entre la intranet, el Internet y varios servicios proxy en el gateway (por lo general este gateway es multihomed).

En el ejemplo anterior el equipo A se hace una consulta a WWW al equipo G. Luego G hace una consulta al Internet y toma la respuesta (proxies). Luego G sirve la respuesta a A.

El gateway toma consultas ingresadas desde el intranet y transmite entonces al Internet usando su propia dirección IP (193.13.5.3- clase C). recibe la respuesta y la toma localmente, ej; al servicio de proxies. La consulta del intranet entonces puede ser satisfecha desde el cache local y las máquinas del intranet no necesitan estar conectadas directamente al Internet.

En el ejemplo anterior, la máquina gateway esta además ejecutando las funciones de firewall (sin permitir ciertos tipos de conexiones) para proteger la intranet de ataques foráneos desde afuera.

Del mismo modo en el ejemplo anterior, la organización sólo necesita una única dirección IP para ser conectada completamente al Internet. En practica, muchos son usados para resistencia y redundancia (un número de gateways), la organización lo más seguro que es dueña de por lo menos una dirección Class C para ser usada con Internet.

**Ejercicio 2-5: Direccionamiento de IP**

En este ejercicio, vamos a ver las direcciones IP de la red de la Compañía X. No hay soluciones para este ejercicio.

**Red de la Compañía X**

El administrador de la red de X ahora necesita decidir en un nuevo esquema de direcciones IP para la redes nuevas de la compañía. Se le pide una propuesta para la compañía.

Las necedades expresadas por los departamentos incluyen:

• Trabajadores en la sede principal, realmente sólo necesitan acceso al sistema de la red interna de la Compañía X más la habilidad de navegar el Internet.

• El equipo de desarrollo necesitan acceso completo al Internet.

• El equipo de ingenieros esta preparado para considerar cualquier propuesta pero están preocupados acerca de costos de equipos y quien va a dar soporte a toda la infraestructura de la red.

Preguntas a considerar:

1.¿Debe X usar una red de clase A, B o C o múltiples redes de una clase en particular?

2.¿Cómo se asignaran números de redes a las redes en su diseño?

3-¿Debe X considerar establecer una intranet para ahorrarse direcciones IP?

4. ¿Qué otras opiniones son arrojas de estas consideraciones?

**Sub-Redes**

Las Sub-redes son una forma muy útil de organizar hosts grupos lógicos dentro de una red. De este modo, una red puede ser dividida en muchas sub-redes. Muchas compañías tienen una sub-red para cada departamento en su organización. Las sub-redes son además útiles cuando los Estándares de la red limitan la habilidad para la red crecer. Por ejemplo, una red ethernet 10BaseT permite un segmento de una longitud de no más de 100 metros, o 328 ft. Para extender la red, podemos crear muchas sub-redes desde la dirección de red existente y conectar cada nodo de la sub-red a un router. Luego, configurar el router para enviar paquetes entre las sub-redes.

El enrutamiento de sub-red permite que numerosas sub-redes existan dentro de una red. Los bits del host son divididos en dos grupos: sub-red y host, por ejemplo, subnetear pide prestado de los 16 bits más bajo (los bits del host) para una red Clase B y los 8 bits para una red Clase C.

Por consiguiente, una dirección de sub-red consiste de las siguientes tres porciones:

**Porción de la red | Porción de la sub-red | Porción del host**

La única forma de identificar la red, sub-red y la porción de host de una dirección IP es introduciendo un segundo elemento llamado "mascara de sub-red". La mascara de sub-red es un elemento obligatorio del TCP/IP. Siempre esta configurada con una dirección IP; estas trabajan en pareja en el sistema. La dirección IP en sistema y la mascara de sub-red es el requerimiento mínimo para una configuración TCP/IP.

## Mascaras de Sub-redes

Una mascara de sub-red llamada además "netmask" ("mascara" como la llamaremos de ahora en adelante), es un número de 32-bit (similar a una dirección IP) con una correspondencia uno-a-uno entre cada uno de los 32 bits en la dirección de Internet.

Las mascaras de Sub-red sirven dos propósitos principales:

- Para distinguir la porción de la red y el host de una dirección IP.
- Para especificar si una dirección de destino es local o remota.

### *Distinguiendo las Porciones de Red y Host de una Dirección IP*

Las mascaras de sub-red distinguen las porciones de red y host de una dirección IP. Porque el sistema no conoce Cuáles son los bits en el campo de los host a ser interpretados como una parte de la subred de la dirección de Internet, el sistema consulta la mascara de sub-red. La mascara de sub-red le dice al sistema Cuáles bits de la dirección de Internet deben ser interpretados como la red, sub-red y dirección de host.

El tipo más simple de mascara de sub-red es la mascara sub-red por defecto. Por defecto, cada campo de 8-bit es encendido (255- todos los unos binarios) o apagado (0- todos los ceros binarios), dependiendo de la clase de dirección (A, B o C).

La siguiente lista identifica las mascaras de sub-red por defecto para las direcciones de clase A, B y C. Las clases de direcciones D y E no tienen hosts y por tanto no requieren mascaras de sub-red.

- 255.0.0.0 Clase A (defecto)
- 255.255.0.0 Clase B (defecto)
- 255.255.255.0 Class C (defecto)

### *Especificando si una Dirección de Destino es Local o Remota*

Las mascaras de subred especifican si la dirección de destino es local o remota. Note que la mascara de sub-red es usada para enmascarar la dirección de red así solo la dirección del host permanece. En enrutamiento, esto es extremadamente importante. Le permite a un computador determinar si el destino de una dirección esta en una misma computadora (local) o una red diferente (remota).

Si la dirección de destino esta en la misma red, la información puede ser transmitida localmente. Si la dirección de destino esta en una red diferente, la información tiene que ser enviada a un router, el cual puede localizar la red remota.

### Ejercicio 2-6: Determinar las Mascaras de Sub-red por Defecto

En este ejercicio, debes determinar la mascara de sub-red por defecto de cada dirección IP. Las soluciones para este ejercicio están localizadas en el Apéndice B.

1. 17.223.13.222
2. 194.10.99.2
3. 211.34.126.10
4. 152.4.202.69
5. 128.156.88.1

### Interfases de Red

Hay varios tipos diferentes de interfases de red utilizadas para conectarse a una red. Algunos tipos comunes de interfases incluyen la interfaz loopback, Ethernet, Token Ring, SLIP, PPP y plip. Puede haber más de una interfaz para cada tipo listado, en cualquier caso, las interfases son enumeradas consecutivamente iniciando con 0. Por ejemplo, eth0 designa la primera interfaz Ethernet.

Los siguientes tópicos son discutidos en esta sección:

- Nombres de Interfases
- ifconfig
- Alias de Interfases
- Configuración Dial-Up
- Configuración de la Red
- Scripts de Inicio de Red
- Herramientas de Configuración
- Probando Conectividad con ping

### Nombres de Interfases

En esta sección, cubriremos algunas de las interfases de redes más comunes.

#### *lo*

La interfaz lo de red es conocida como la interfaz loopback. El propósito de lo es direccionar el tráfico que se origina desde la máquina local que además tiene la máquina local como destino final. La dirección loopback tiene una dirección IP especial que esta asignada para tráfico local. Esta dirección es 127.0.0.1. Todo computador ejecutando GNU/Linux necesita tener una interfaz lo configurada y activa.

#### *eth*

La interfase eth es usada para conectarse a una red ethernet. Las redes ethernet son los tipos más comunes de interfases usadas para conectarse a una LAN debido a que ellas alto nivel de transmisión de información, permiten múltiples equipos, y son fáciles de implementar. Si tienes una tarjeta de red, lo más probable que esta sea un tarjeta ethernet.

***tr***

Una interfaz tr designa una interfaz para un Token Ring. Las redes Token Ring son muy útiles ya que ellas evitan el problema de colisiones de data de las Ethernet. Este tipo de LAN no es tan común como el Ethernet ya que las redes Token Ring no son tan fáciles para implementar.

***SLIP***

Las interfases SLIP son interfases de líneas seriales. Las interfases seriales son comúnmente usadas para conectar computadoras aisladas a otras redes de computadoras mediante un modem, pero ellas pueden además ser usadas para conectar dos máquinas directamente. Un problema que existe cuando se usan líneas seriales para conectar a una red es que la transmisión serial de información no es muy rápida debido a limitaciones de equipo (hardware).

***PPP***

Otra interfase de línea serial es la ppp a excepción que usa el protocolo de punto a punto (ppp). Una interfaz PPP es creada cuando un enlace (link) serial es establecido usando PPP. Este tipo de interfaz puede ser usado para los mismos propósitos como la interfaz SLIP, pero una interfaz PPP es mucho más confiable que una interfaz SLIP y su uso es más popular.

***plip***

Las interfases plip son similares a las interfases SLIP a excepción de que ellas son ejecutadas sobre líneas paralelas. El uso típico de una interfaz plip es para conectar dos máquinas por medio de puertos paralelos. Una interfaz plip puede transferir información más rápido que una interfaz de línea serial, pero los periféricos adheridos a el puerto paralelo no pueden ser usados mientras la interfaz plip este activa. Por ejemplo, activando la primera interfaz plip desactivaría lp0 en favor de plip0. Debe notarse que una interfaz plip no puede ser inicializada con un cable paralelo ordinario, se necesita un cable especial llamado Null Printer.

**ifconfig**

El comando ifconfig es usado para configurar las interfases de red en el sistema, ifconfig hace dos funciones básicas. Este puede listar interfases y puede ser usado para configurar más interfases. Las interfases que están activas comúnmente en el sistema pueden ser vistas escribiendo el siguiente comando:

```
# ifconfig
```

El estado de una interfaz en específico puede ser observado al invocar ifconfig con el nombre de la interfaz como el único argumento.

```
# ifconfig eth3
```

Para hacer algo más que desplegar las interfases con ifconfig, el usuario debe tener privilegios de root. El uso más común de ifconfig es para activar y desactivar una interfaz. Esto se hace escribiendo “ifconfig” seguido por el nombre de la interfaz tal como eth2, luego “up” o “down”, up específica que interfaz debería ser activada, mientras que down indica que la interfaz debería ser desactivada. Por ejemplo, el siguiente comando:

```
# ifconfig eth2 up
```

Este comando activa la segunda interfaz ethernet.

Una dirección IP puede ser asignada a una interfaz usando el comando ifconfig. Cuando se usa de esta forma la estructura del comando ifconfig es:

```
#ifconfig [dispositvo] [direccionIP] netmask [número netmask] [up\down]
```

Un ejemplo de configuración de una dirección IP y activar una interfaz es:

```
# ifconfig eth0 192.168.2.7 netmask 255.255.255.0 up
```

En este ejemplo, la interfaz eth0 es configurada con la dirección IP 192.168.2.7 y con una mascara de 255.255.255.0. El uso de up en el comando activa la interfaz. Esta opción puede ser omitida porque es la acción por defecto. Para desactivar la interfaz de red se escribe:

```
# ifconfig eth0 down
```

Mientras se configuran las interfases de red, el kernel asume ciertos valores por defecto. En el ejemplo anterior, han sido especificadas la dirección de red y la mascara. Si alguna de estas direcciones no hubiese sido proveída, el kernel tomaría unos valores por defecto basados en la información que si fué suplida. Si la mascara no fué suplida, el kernel lo asumiría como que es una red Clase C que se está configurando y la configuraría como una dirección de red de 192.168.2.0 y una dirección de broadcast de 192.168.2.255 para la interfaz.

Hay muchas opciones de línea de comando para controlar la funcionalidad de una interfaz. Algunas de las opciones del comando ifconfig son las siguientes:

| | |
|---|---|
| **up** | Esta opción activa la interfaz. |
| **down** | Esta opción desactiva la interfaz. |
| **[-]arpa** | Esta opción activa o desactiva el uso de arpa en la interfaz. |
| **[-] promisc** | Esta opción activa o desactiva el modo promiscuo de la interfaz. Si esta especificada, todos los paquetes en la red pueden ser recibidos por la interfaz. |
| **[-]allmulti** | Esta opción activa o desactiva la recepción de todos los hardware multicast packets. |
| **mtu N** | Esta opción le permite configurar la máxima Unidad de Transacción (Maximum Transaction Unit (MTU)) de este dispositivo. |
| **netmask <addr>** | Esta opción permite al usuario configurar la mascara de red perteneciente al dispositivo. |
| **[-] broadcast [addr]** | Esta opción nos permite activar, desactivar y configurar la aceptación de datagramas para la dirección broadcast específica. |
| **[-] pointopoint [addr]** | Esta opción nos permite configurar la dirección de las otras máquinas al final de la longitud del punto a punto, tales como slip o ppp. |

Además están disponibles opciones avanzadas para controlar el comportamiento de la interfaz, tales como las configuraciones de hardware, asignación de protocolos a la interfaz y opciones para configurar una interfaz para usar IPv6.

### Alias a Interfases

Es posible asignar múltiples direcciones IP a una misma interfaz. Este proceso es conocido como crear alias de IP. Un beneficio de tener una dirección IP asignada un alias es que le permite múltiples direcciones IP apunten a una misma localidad Física. Esto puede ser muy útil para establecer un host en dos redes separadas sin la necesidad de dos dispositivos de red. Uno puede además querer asignar a un único host múltiples direcciones IP y así poder ejecutar múltiples servidores pero con direcciones diferentes, está técnica es que hoy día hace posible tales como los servidores Web.

Tradicionalmente, un host multihomed es uno con múltiples interfases. Cada interfaz debe tener una dirección IP distinta. Esta interfaz puede ser de cualquier forma o conexión tal como una tarjeta Ethernet, DSL, un vinculo punto-a-punto o un cable modem. De este modo un host que tenga múltiples interfases sería una máquina con un vínculo Ethernet y uno punto-a-punto. Un enrutador es un dispositivo multihomed porque envía datagramas que llegan en una interfaz y entregan en una interfaz diferente. Un host multihomed no es un router a menos que remita los paquetes. Cuando se cuenta el número de interfases para ver si un host es multihomed, la interfaz loopback no es incluida. Esta forma de multihoming ha pasado a ser obsoleta y una nueva generación de multihoming se ha popularizado.

Una interfaz alias es similar a una interfaz regular excepto que otro número es adherido al nombre de la interfaz para designar el número alias. La interfaz eth0:0 es la designada primer alias de eth0. El interfaz eth0 es la misma interfaz física que eth0:0 excepto con una dirección IP diferente. Para que los alias funcionen

después de reiniciar, los pasos de la configuración deben colocarse en un shell script que se ejecutará cuando se inicie el sistema.

Para configurar un IP alias, el kernel debe tener soporte de alias de IP, ya sea dentro del kernel o como un módulo. El siguiente comando puede ser usado para cargar el módulo de alias de IP:

```
# /sbin/modprobe ip_alias
```

La interfaz que se le va a aplicar el alias debe existir en el sistema. Por ejemplo, antes de que se pueda crear eth0:0, la interfaz eth0 debe existir. Esto se puede hacer mediante las herramientas de configuración de las distribuciones específicas o se puede hacer manualmente. Los permisos de Root deben estar disponibles para permitir estas acciones.

Primero, la interfaz debe ser asignada su propia dirección IP con ifconfig. Una vez la interfaz obtenga el número de IP, esta empezará a interactuar con la red. Cuando la interfaz este lista y funcionando se debe usar ifconfig nuevamente para asignar los números IP alias. Con un comando similar al siguiente:

```
# ifconfig devx:y dirección-ip up
```

creará el alias y para el dispositivo m, asignando la apropiada dirección IP, entonces funciona la interfaz. Por ejemplo:

```
# ifconfig eth0:0 152.168.1.20 up
```

Iniciaría activado un alias de eth0 con el la dirección IP de 192.168.1.20.

Aunque este ejemplo solo muestra la adicción de un alias, cada interfaz de red puede por defecto aguantar un máximo de 256 alias. El número por defecto esta definido en /usr/include/linux/net_alias.h.

Para borrar un alias, se usa la siguiente línea:

```
# ifconfig eth0:0- 0
```

El espacio después del eth0:0 indica eliminar el alias y esto se puede hacer para cualquier alias. Para guardar los cambios (alias creados y/o borrados) al reiniciar, hay que agregar las líneas para iniciar o matar los alias a un script que se ejecute cuando el sistema se inicia, tal como el archivo /etc/rc.d/rc.local.

## Configuración Dial-Up

Para sistemas que no estén conectados a una red, muchas veces la única forma para conectar a otras máquinas es mediante una conexión serial, como es un modem. El protocolo usado cuando la red esta mediante líneas seriales es el Protocolo Punto-a-Punto (Point-to-Point Protocol (PPP)). Cuando se inicia una conexión serial, pppd es usado para establecer la conexión. Para conexiones seriales sobre líneas ISDN, se usa una versión especial de pppd, ipppd.

### *Demonio del Protocolo Punto-a-Punto (Daemon pppd)*

Mientras TCP/IP y otros protocolos son más comunes en el Internet, PPP es aún el protocolo usado para la conexión dial-up. GNU/Linux tiene un demonio que interactua con el resto de el Internet mediante PPP. El pppd tomará los paquetes desde una red y lo convertirá entonces en un formato aplicable para transmisiones PPP. Este demonio esta ubicado en /usr/sbin/pppd. Hay muchas formas para iniciar un demonio. Puede ser iniciado manualmente desde la línea de comandos con scripts o usando un programa por separado para hacer las conexiones, como era el dip. El dip (Dial-Up IP Protocol Driver) puede ser usado como un programa automatizado o un utilitario interactivo para conectar el modem a una conexión e iniciar el pppd. Otras formas es usar los utilitarios gppp y kppp para conectar automáticamente.

Para configurar y subir pppd en una máquina, hay dos partes a configurar. Primero, debe configurar PPP en el kernel. Esto es normalmente una opción por defecto cuando el kernel es inicialmente configurado. Para

revisar, debe observar el mensaje al iniciar y observar las líneas que contienen información acerca del PPP. Si PPP no esta configurado en el kernel deberá hacerlo antes para que el pppd trabaje. Para configurar e instalar el pppd, puede usar un archivo RPM o un tar.gz de fuente. La información importante que debe conocer para la instalación es el tipo de modem, velocidad y cual dispositivo este usa en GNU/Linux.

El pppd debe estar ejecutándose para poder usar el PPP. El PPP es un sistema en tres partes y necesita el pppd para controlar cada aspecto de el PPP. Las tres partes del PPP son un formato estándar de data para enviar información a través líneas seriales, un Protocolo de Control de Vínculos (Link Control Protocol (LCP)) y una colecciones de Protocolos de Control de Red (Network Control Protocols (NCPs)). El formato estándar usado para enviar información es una trama. Las tramas son paquetes de data que contienen un bloque cabezal (el cual identifica la trama) y un block de data (la información actual que necesita ser transmitida). El LCP es usado para transportar las tramas que controlan el PPP y establecen, mantienen y terminan la conexión. NCP es el protocolo de configuración para los protocolos de la capa de Red. Hay un NCP individual para cada protocolo de la capa de Red que es soportado por PPP.

Para usar pppd, debe haber información disponible para el demonio. Algunos de los diferentes tipos de información que el demonio necesita saber incluyen el tipo de modem usado, si usan y que tipo de autentificación esta presente y cual NCP usan (por lo general es el IP Control Protocol (IPCP)). El pppd es bastante versátil, con muchas opciones que pueden ser llamadas al inicio del demonio. Además de las opciones configurables listadas previamente, algunas opciones incluyen la velocidad de conexión, donde se encuentra el script de conexión, si usar una ruta dedicada mediante las tablas de enrutamiento una vez que la conexión este establecida, si se necesita cerrar (lock) el dispositivo en uso por el demonio, que tan grandes pueden ser los paquetes recibidos (Maximum Receive Unit (MRU)]) y que tan grandes pueden ser los paquetes enviados (MTU).

El MRU específica el tamaño de la trama más grandes que pueden ser recibidas en una conexión. El MTU es el tamaño más grande de las tramas que pueden ser enviadas en una conexión. La medida por defecto para ambos casos es 1,500 bytes. Una medida alternativa es de 576. Estas medidas son negociables al inicio de una sesión entre dos puntos finales de la conexión.

El archivo de configuración /etc/ppp/options es usado por todo el sistema para la configuración del pppd. Un ejemplo de un archivo de /etc/ppp/opcions es:

```
$  Log PPP control   framss  to syslog
debug
1 Serial  interfase options
/dev/ttysl  33600
lock
modem
crtscts
$  User name  for PPP authentication
user myaccount
remotename itiyisp
$ Use PPP peer as default gateway in system routing  table
defaultroute
```

Este archivo de configuración muestra un usuario quien esta marcando a través de la interfaz serial /dev/ttys1 a 33600 bits por segundo. El control de hardware (crtscts) y el control de modem (modem) están habilitados. La opción de la ruta por defecto (defaultroute) establece que el otro punto de la conexión PPP es el gateway por defecto para todo tráfico saliente. Alguna de las otras opciones que son posibles incluir en el archivo son: asyncmap, auth, connect, lock, idle y ktune. La opción asyncmap específica Cuáles caracteres necesitan ser escapados para una transmisión correcta. La opción auth requiere que el compañero (peer) se

autentifique antes de que los paquetes puedan ser aceptados o enviados. Connect es una opción que dice Cuáles scripts usar para configurar una línea serial. Lock, cuando se habilita, puede bloquear el dispositivo serial, así que solo el pppd lo puede usar. Idle puede especificar el número de segundos suspendido (idle) permitidos antes de que se efectúe una desconexión automática. La opción ktune es usada para permitir a pppd cambiar el kernel como sea necesario mientras se ejecuta.

Cuando se inicia el pppd, solo los errores fatales se muestran en pantalla. Los errores que no son fatales se almacenan en el archivo /var/log/daemon.

Aunque PPP es usado principalmente sobre conexiones dial-up, es posible usar pppd sobre líneas permanentes. Un demonio separado se ha escrito para usar PPP sobre ISDN, llamado ipppd.

**Demonio de Protocolo Punto-a-Punto de ISDN (ipppd)**

Para líneas ISDN, hay un demonio separado que es usado que no es pppd. El demonio ipppd es para sistemas GNU/Linux que se conectan sobre ISDN. De cualquier manera el usuario tiene que decidir si usa una conexión PPP sincrónica o asincrónica. Si el PPP en uso es asincrónica, lo mejor es usar el pppd normal con dispositivos ISDN y no el ipppd. Los emuladores existen para probar dispositivos que corren sobre líneas ISDN para transmisiones asincrónica que trabajan con el pppd estándar.

Igual que el pppd, ipppd tiene un archivo de configuración global: /etc/ppp/ioptions. Este archivo puede permitir al administrador controlar las opciones que son aceptadas por ipppd. Muchas de las mismas opciones de pppd son además usadas por ipppd, tal como nombre de dispositivo(s), si usar autentificación y si el dispositivo(s) tiene que ser bloqueado exclusivamente para el uso del ipppd. De cualquier manera, es importante recordar que algunas opciones que so necesarias para el pppd no lo son en una conexión ISDN. Esto incluye velocidad de transmisión, que hacer después de una desconexión y cualquier manejo de transmisiones asincrónica.

El ipppd puede manejar más que un dispositivo que este conectado mediante líneas ISDN. Pero, el ipppd es solo iniciado una vez, sin importar cuantas conexiones se están ejecutando.

El ipppd además configura y transmite LCP y NCP para las transmisiones. El LCP es usado para transportar las tramas de control (paquetes de datos) para la conexión. Los NCPs son los protocolos para las diferentes protocolos de la capa de Red que pueden estar presente en una conexión PPP.

**Configuración de Red**

Hay más en configurar en una red que activar una interfaz y asignarle una dirección IP. La resolución de nombres y enrutamiento son exactamente dos de muchos componentes necesarios que conforman el funcionamiento y uso de una red.

***Enrutamiento***

Después de que el dispositivo ha sido asignado su dirección IP y ha sido puesto a funcionar, el comando route debe ser usado para establecer el enrutamiento para la nueva interfaz si se esta usando enrutamiento estático. El comando exacto para configurar una ruta por defecto puede variar dependiendo en la configuración específica de la red, pero la forma general del comando se muestra a continuación.

```
# route add  opciones
```

Las opciones varían dependiendo en la configuración particular de la red a la que el sistema este conectado, pero las partes route y add del comando son requeridas. Algunas opciones que pueden ser necesarias son el gw para establecer el gateway por defecto, -net para establece una ruta a una red o -host para establecer una ruta a otro host. Por ejemplo:

**# route add default gw 192.168.2.7**

significaría que todos los paquetes destinados para el exterior de la red puedan usar 192.168.2.7 como el gateway por defecto.

### Resolución de Nombre

Para identificar una interfaz de red, un servidor de nombre (name server) debe establecerse. Este asignará un nombre de dominio a la dirección IP y es establecido editando el archivo /etc/resolv.conf. Un ejemplo de que debe aparecer en el archivo es similar a las siguientes dos líneas:

```
domain abiertos.org
nameserver 192.168.2.7
```

La línea domain designa específicamente el nombre del dominio del sistema local. La segunda línea especifica la dirección IP del servidor de nombre primario que debe ser usado. Múltiples servidores de nombres pueden ser usados en caso de emergencia que el servidor primario no pueda ser accesado.

El comportamiento de la resolución de nombre puede ser afectado aún más editando el archivo /etc/host.conf. Normalmente este archivo contiene las siguientes dos líneas para permitir un archivo hosts local:

```
order hosts.bind
multi   on
```

La primera línea es este archivo dice que cuando se busca un nombre, el archivo de los hosts locales es revisado primero y entonces el Sistema de Nombre de Dominio (Domain Name System (DNS)) es revisado de acuerdo con las instrucciones en el archivo /etc/resolv.conf.

El siguiente paso en manualmente configurar la interfaz de red implica editar el archivo /etc/hosts. Este archivo contiene ambos, los nombres y las direcciones IP de todos los hosts locales. Una vez que un host es colocado en este archivo, el servidor de nombre de dominio no tiene que ser consultado para obtener su dirección IP. Pero recuerde que, este archivo debe ser actualizado en caso de cambio de una dirección IP a uno de los hosts. Si el sistema es bien administrado, los únicos nombres de host que deben aparecer en este archivo son interfaz loopback, y el nombre del host local. Una entrada típica en este archivo es:

```
# <Dirección IP>        <nombre host>          <alias>
192.168.2.2             eq2.abiertos.org       dos
192.168.2.7             eq7.abiertos.org proxy siete
```

Todas las líneas anteriores inician con la dirección IP, luego la segunda parte es el nombre del host, seguido por un alias (si alguno) para ese host. Generalmente, los alias no son requeridos, pero la única ventaja de tenerlos es que las máquinas pueden ser accesadas en la red local escribiendo el nombre alias cortos en lugar de el nombre completo del host. En el ejemplo, eq7.abiertos.org puede ser accesado localmente como proxy.

### Scripts de Inicio de la Red

Puesto que hay muchos pasos para establecer y configurar una interfaz de red, algunas distribuciones han tomado pasos para facilitar la configuración de interfases de red de todo tipo. Estos scripts son más útiles cuando se usan interfases como las de ppp que necesitan ser actividades y desactivadas con más frecuencia. Estos scripts se encuentran en el directorio /etc/sysconfig/network-scripts.

El directorio /etc/sysconfig/network-scripts contiene tres diferentes tipos de archivos: archivos de configuración, archivos de activación y archivos de desactivación. Los archivos de configuración son de la forma ifcfg-interfase y contienen información como son la dirección IP, la dirección y mascara de red. Los archivos de configuración pueden además contener parámetros para activar una interfaz en el inicio y algunos otros parámetros específicos a la interfaz.

Los archivos de activación siguen la convención de nombre ifup-interfase y contienen código para activar la interfaz específica. Los archivos de desactivación son usados para desactivar una interfaz y son de nombres parecidos a: ifdown-interfase.

Además de los tres tipos de archivos previamente listados, hay otros archivos especiales para asistir las configuraciones de red. El más importante de estos archivos es network-functions. Este archivo no es un shell script y contiene cierta importantes informaciones para los scripts. Otro script especial es ifup-aliases y es usado para configurar alias para interfases en el sistema. Antes que una interfaz pueda ser asignada alias, la interfaz debe existir físicamente, estar configurada y activa. El último shell script es ifup-routes y es usado para configurar enrutamiento estático para el sistema. Para ifup-routes ser útil, la red tiene que estar emplear enrutamiento estático.

Cuando se usan estos dos últimos scripts (ifup-aliases/ifup-routes), es importante notar que ellos solo pueden ser usados en sistemas Red Hat desde que ellos hacen llamadas a programas de administración específico a Red Hat. Ellos además, deberán ser ejecutados como root ya que ellos requieren los privilegios del superusuario para ejecutar estos programas de administración. Ejecutarlos con el comando sudo puede resolver este problema.

Una manera de aprovecharse de los scripts de inicio de red es usando los comandos ifup e ifdown. Estos scripts, ubicados en /sbin, pueden ser usados para activar o desactivar una interfaz. Estos permiten al sistema mantener las configuraciones actualizadas sin la asistencia del administrador. Los cambios pueden ser configurados y almacenados usando utilitarios de administración tal como netcfg y network-admin de gnome. A veces es buena practica en el sistema permitir a ciertos usuarios, que no sean administradores, activar y desactivar algunas interfases, especialmente interfases ppp y SLIP. Por ejemplo, podemos usar el siguiente comando para inicializar una conexión PPP:

```
$ /sbin/ifup ppp0
```

**Herramientas de Configuración**

Hay un número de herramientas disponible para configurar las interfases de red. El único común de todas las distribuciones GNU/Linux y las mayoría de variantes de UNiX es ifconfig, una herramienta de línea de comando. La mayor parte de las grandes distribuciones de GNU/Linux tienen además sus propios programas gráficos o basados en menu de configuración. Los dos desktops más usados también proveen interfases gráficas para la administración de los dispositivos de red de todo tipo. GNOME usa network-admin y KDE tiene a Kcontrol estos dos tools hacen a los archivos y producen configuraciones más fácil luego de mantener. Pero si alguien eligió trabajar en MANDRIVA, SuSE, RedHat o cualquier de las otras grandes distros de GNU/Linux debe acojerse a sus herramientas de configuración ya que esto es de las principales razones que ellas ofrecen para elegirlas como plataforma de trabajo.

Aquí en esta sección discutiremos las diferentes herramientas de configuración de interfases de red proveídas por las principales distros. No es en ninguna manera la intensión de cubrir todos los aspectos o de ser completo a todas distros y a toda herramienta.

**YaST2 (SuSE)**

Yet Another Setup Tool 2 es una herramienta basada tanto en GUI (Interfases Gráficas de Usuarios) o MDUI (Interfases de Usuarios Manejadas por Menus) manejado a través de Menus que es la parte integral de la distribución SuSE GNU/Linux. YaST2 es la herramienta principal para la instalación, configuración y administración de hosts ejecutando SuSE. Esta sección trataremos sólo con la capacidades de configuración de red YaST2.

Para accesar a YaST2, hay que iniciar en el host como root y seleccionar YaST2 en el menu de KDE o

desde un terminal. Cuando la ventana de YaST2 aparece, el usuario puede seleccionar la administración del sistema desde el menu principal y luego la configuración de la red en la ventana pop-up que aparece. Esto permite hacer la configuración a los host de la red. El usuario se le presentan muchas opciones, sin embargo sólo unas cuantas pertenecen directamente a la configuración de la interfaz de la red.

La configuración base de la red es donde las direcciones IP son configuradas para las interfases de red identificadas en el sistema. Aquí se les puede asignar direcciones IP a los dispositivos de red y pueden ser activados o desactivados. Aquí en este punto, podemos elegir que tipo de red estamos configurando (tal como Ethernet). La opción cambiar el nombre del host es usada para configurar el nombre del host así como el nombre del domino al que el host pertenece.

La opción de cliente DHCP se encontrará activada sólo si el paquete dhclient ha sido previamente instalado. DHCP es usado si el host recibe las direcciones IP y la información de enrutamiento dinámicamente en la red a la que este conectada.

La opción de configurar parámetros ISDN esta disponible solo si el paquete i4l ha sido instalado. Esta opción permite a los usuarios configurar las opciones que su tarjeta ISDN usa para conectarse al proveedor de servicios. Esta opción fija la configuración para elementos tales como tipo de red, host y números de teléfono remotos, número de llamadas recuperadas, tiempo de espera entre las llamadas y login/clave.

**Netconfig (Red Hat/Fedora)**

Los usuarios de la distribución Red Hat GNU/Linux tienen el programa netconfig para ayudar en la configuración de una red. Netconfig puede ser usado para configurar muchas características de la red, incluyendo ambas tareas cliente y servidor.

El propósito del utilitario netconfig es permitir a usuarios configurar su red TCP/IP desde cero. Netconfig provee un GUI a través del cual casi todos los aspectos de los sistemas de la red pueden ser definidos y controlados. Los usuarios son permitidos controlar casi todas las facetas de la red de su computador desde la información del host y enrutamiento hasta Apache y Samba.

Primero, para que los usuarios puedan tener acceso a netconf, los usuarios deben tener privilegios de root. No es buena práctica permitir a cualquier usuario en un sistema editar o controlar funciones básicas de la red, puesto que podrán interrumpir fácilmente el sistema de la red. Una vez que el usuario ha accesado como root, netconf es muy fácil de iniciar. Este se encuentra en el directorio /usr/sbin y puede ser lanzado simplemente escribiendo en la línea de comandos la siguiente sentencia:

```
# /usr/sbin/netconfig
```

Después de que netconfig ha sido iniciado, la ventana de netconf se presenta en el escritorio. Dentro de la ventana principal se encuentran tres tabuladores: tareas de cliente, tareas de servidor y miscelaneo.

***Tareas del Cliente***

La sección de tareas de cliente de netconf contiene las herramientas necesarias para configurar las opciones de red del host local. La ventana Información Básica del Host permite al usuario entrar el nombre del host de la computadora y configurar cualquier adaptador que este adherido al sistema.

Especificación de Nombre del Servidor es usada para especificar el nombre del dominio local, nombre de servidores usados por el host local y en Cuáles dominios buscar cuando el nombre de un sistema este pasado como local (un nombre que no incluya un dominio).

Las opciones de Enrutamiento y Gateways toma incontables selecciones, todas de las Cuáles permite a un usuario configurar diferentes rutas y gateways a otras redes o hosts. Esta parte del utilitario además otor-

ga acceso a la configuración del demonio routed, el cual es usado para redes complejas que interconectan grandes redes.

La Ruta de Búsqueda de Nombres de Host (Host Name Search Path) configura como los diferentes servicios de nombres serán probados por el host. Las tres opciones son hosts para el archivo /etc/host, NIS para usar el Network Information System y DNS para usar el Sistema de Nombre de Dominio. Los tres nombres de servicios pueden ser probados en cualquier orden y es posible seleccionar una configuración donde uno o dos de los servicios no sean probados completamente.

La sección de Ajuste de Interfaz IPX permite configurar el host para usarse en una red IPX, mientras el botón PPP/SLIP/PLIP hace lo mismo para las redes PPP/SLIP/PLIP.

***Tareas de Servidor***

La sección (Server Task) de Tareas del Servidor de netconfig es para configurar el host local para usarse como un Servidor en la red. a través de la asignación de alias de IP para hosts virtuales, es posible asignar múltiples direcciones IP a un dispositivo de red. Esta es la sección donde los alias son configurados.

***Misceláneos***

Esta última sección principal tiene dos partes importantes: información acerca de otros hosts e información acerca de otras redes. La información acerca de otros hosts y la información acerca de otras utilidades de redes muestran la información en los archivos /etc/hosts y /etc/networks, respectivamente. Ellos permiten modificaciones de entradas existentes así como la adicción de nuevas entradas.

**Netconfig (SlackWare)**

Contrario a las tres otras herramientas discutidas en esta sección, netconfig es realmente un script que esta diseñado para ayudar al usuario novato en la configuración e instalación de los archivos de configuración de TCP/IP necesarios para conectar el host a una red. Netconfig le hace una serie de preguntas al usuario y luego crea el archivo rc.inet1. Este archivo es usado por el host para configurar los dispositivos de red adheridos al sistema y enrutando. Netconfig es además capáz de probar y activar la tarjeta de red para el sistema.

Netconfig es iniciado escribiendo /sbin/netconfig en la línea de comandos. Este sólo puede ser llevado acabo por el super usuario root porque netconfig hace cambios a archivos de configuración del sistema.

Una vez que netconfig esta ejecutándose, la primera información que deberá suplir es el nombre del host. Este es el nombre de la máquina que netconfig se está preparando para configurar. La entrada no incluye el nombre de dominio del host. El nombre de dominio es pedido próximo y no incluye el (punto) delantero ".".

Luego, netconfig puede preguntar si el usuario desea utilizar TCP/IP a través del loopback. Esto solo se puede hacer si el usuario va ha usar el host únicamente como una máquina aislada (stand-alone). Si el usuario quiere usar el host en una red, la respuesta a esta pregunta debe ser no.

La siguiente serie de preguntas son acerca de números IP que serán usados por el host para comunicarse en la red apropiadamente. La primera es la dirección IP para la máquina local, la segunda es la dirección para el gateway y la tercera es la mascara de red. Toda esta información se necesita para que el host trabaje apropiadamente en la red.

Finalmente, netconfig le preguntará al usuario que servidor de nombre será usado. Si el usuario responde afirmativamente, entonces netconfig le permite al usuario entrar la dirección de servidor de nombre. Si necesita ingresar un segundo y tercer servidor de nombres, el usuario puede adherirlos editando el archivo /etc/resolv.conf.

Los archivos de configuración /etc/rc.d/rc.inet1 y /etc/rc.d/rc.inet2 pueden ser modificados cuando sea para cambiar la configuración de la red. Como se estableció previamente, el archivo rc.inet1 configura los dispositivos de red y configura el enrutamiento. El archivo rc.inet2 no hace el lanzamiento actual de aplicaciones y demonios que corren el la red, incluyendo el demonio inetd que ejecuta en la mayor parte de los otros servicios de red.

**WvDial**

WvDial es una aplicación diseñada para ayudar a configurar un sistema GNU/Linux para usar una conexión dial-up para el Internet. WvDial es útil para los usuarios porque puede detectar el modem y sólo requiere del usuario colocar el login de usuario, la clave y el número de teléfono que se necesita para conectarse. Una característica de WvDial es que no requiere de ningún script.

Se necesitan algunos pasos para configurar y ejecutar WvDial. Primero, editar el archivo wvdial.conf localizado en /etc/. En el archivo wvdial.conf, el usuario necesita editar el nombre de usuario, clave y número de teléfono para marcar. Después de editar este archivo, el usuario puede invocar el comando wvdial y la computadora marca al ISP del usuario.

Si se prefiere una aplicación GUI, el usuario puede usar las aplicaciones x-Wvdial o KWvDial que instalan las aplicaciones marcadoras usando una interfaz gráfica. La misma información que se necesita para el programa WvDial de línea de comandos se necesita para estos programas.

Para configurar el dial-up en estas aplicaciones, un usuario debe iniciar en File en el menu estándar, luego New y finalmente responder las preguntas. Una vez acabado, el usuario debe seleccionar conectar y entrar el nombre de usuario, clave y número a marcar.

**Probando la Conectividad con ping**

Una de las herramientas de troubleshooting más poderosas que puede usar cuando se configuran las interfases de red es el comando ping. El comando ping envía un paquete IP de información y espera por una respuesta. Si la respuesta es recibida, ya se sabe que la red y la interfaz están trabajando. Cuando se usa ping para examinar una interfaz, debe ejecutarlo desde la misma sub-red para que los problemas con el enrutamiento no afecten el resultado.

El ping funciona enviando un mensaje ICMP tipo 8 al host especificado. Cuando el destinatario recibe el mensaje, le responde con otro mensaje ICMP tipo 0. Si la red esta trabajando entre los dos hosts, el ping deberá ser retornado. Si hay problemas de red entre los host, el ping no será retornado. El ping puede ser invocado desde la consola con el siguiente comando: **$ ping host.dominio**

**Ejercicio 2-7: ifconfig**

Este ejercicio demuestra el uso del comando ifconfig, el cual es usado para ver y configurar las interfases de red. Debes ingresar como root para completar este ejercicio. Las soluciones para este ejercicio están proveídas en el Apéndice B.

1. Mostrar la configuración actual.
2. Mostrar la configuración actual de la primera tarjeta Ethernet.
3. Escribe debajo tu dirección IP y mascara de red.
4. Desactive la interfaz ethernet primaria.
5. Cambia tu dirección IP a 10.10.10.10, con una mascara de red de 255.255.0.0 y activa la interfaz.

6, Muestra los ajustes en la interfaz para que sea segura y correcta.

7. Adhiera un alias a la interfaz de 10.10.10.9 netmask 255.255.0.0.

8. Visualice la tabla de enrutamiento.

9. Restaura su dirección IP original. Reinicia si tienes problemas para que su conexión a la red trabaje nuevamente.

### Ejercicio 2-8: tcpdump

Este ejercicio da un repaso completo a través del uso del utilitario tcpdump, el cual permite ver los paquetes de la red y como viajan a través de la red. Para hacer este ejercicio, necesitarás iniciar como root y tener la tarjeta de red en modo promiscuo. No se proveen soluciones para este ejercicio.

1. Muestra la configuración actual de tu tarjeta de red*:*

   # ifconfig ethO

2. Ejecute el comando tcpdump sin opciones:

   # tcpdump

3. Observe la información de cada paquete. La mayoría usted no la entenderá, pero podrá descifrar algunas líneas. Si no existe tráfico en la red deberá generarlo desde otra computadora en la subred.

4. Cambiese a otra consola virtual o xterm. Muestra su tarjeta de red de nuevo:

   **# ifconfig eth0**

Esta sentencia le notificará que la interfaz de red tiene la bandera PROMISC activada. Esto es porque tcpdump le dice a la tarjeta de red que quiere ver paquetes destinados para cualquier nodo en la sub-red, no sólo los paquetes direccionados a la dirección MAC de el sistema en el que usted esta.

5. Vuelva a la consola original. Detenga a tcpdump con CONTROL+C. Inicialo otra vez con el siguiente comando:

   **# tcpdump -x -v**

6. Muestre la salida de nuevo. Fijese que hay más información mostrada acerca de cada paquete. La opción -x descarga la cabecera en su código hexadecimal y la opción -v le dice que muestre todo el progreso en pantalla imprimiendo más información acerca de cada paquete. Detenga el programa, después de haber visto varios paquetes.

7- El programa tcpdump viene con un lenguaje de filtrado que se puede usar para mostrar ciertos paquetes. Se puede filtrar por dirección de origen y destino, número de puerto, protocolo, tamaño del paquete y varios otros ajustes. Revise las páginas man para más detalles. Ejecute el comando mostrando solamente los paquetes ICMP:

   **# if tcpdump -x icrap**

8. Efectué un ping a una máquina desde otra máquina en la misma subred:

   **# ping -c 1 sistemaB**

Todo lo que deberá ver es una petición ICMP y una respuesta. El protocolo que el comando ping usa.

9. Use el siguiente comando ping:

   **# ping -p abcd1234 -c 1 sistemaB**

Fijese que usted puede ver el abcd1234 en los paquetes ICMP.

10. Usted puede filtrar exactamente paquetes yendo desde un sistema a otro:
    **# tcpdump src 192.168.0.1 and dst 192.168.0.2**

11. Usted puede hasta filtro solamente tráfico Web:
    **# tcpdump src 192.168.0.1 and dst 192,168.0.2 and port 80**

12. Estudie a ver si puede idear otros filtros interesantes.

13. Si estas ejecutando el X, trate de ejecutar el programa ethereal. El provee un buen GUI para el programa tcpdump. El utiliza la mismas sintaxis de filtrado pero provee más información en un formato que es un poco más fácil para leer.

## EL FUTURO - EL IPV6

Los profesionales del TI tenemos que siempre estar preparado para prever el futuro. Necesitamos una bola de cristal e intentar predecir los acontecimientos, para estimar como los próximos avances tecnológicos van a afectar nuestro trabajo y presupuestos. En este caso, la versión 6 de IP el IPv6, es el futuro y ya esta listo para escribirse.

En esta sección describimos la funcionalidad del protocolo IPv6 de la capa de Red. Se cubrirá el formato del datagrama y cabecera de IPv6, la fragmentación, como sistemas individuales son direccionados usando TCP/IP y algunas de los otros protocolos del TCP/IP que interactúan con IPv6, tales como arpa y el ICMP.

Los siguientes puntos son discutidos en esta sección:
- La necesidad del IPv6
- Repaso del IPv6
- Asignación de Direcciones
- La Transición hacia el IPv6

### La necesidad del IPv6

El crecimiento del Internet es el simple y único factor que motivó la necesidad de desarrollar el IPv6. Las necesidades históricas de suplir espacio de dirección y asignación de direcciones IPv4 no ha resultado lo suficiente para llenar las futuras necedades del Internet.

Debido a este rápido crecimiento del Internet, el enrutamiento se ha convertido poco administrable. El mayor de los obstáculos es el direccionamiento. Las direcciones IP para poder suplir las demandas se están agotando debido al rápido crecimiento y al futuro crecimiento proyectado, y ya que hay un número finito de direcciones, lo que se quiere hacer incrementar el número para el futuro crecimiento del Internet. El IPv6 permite suficiente direcciones, para ser exacto son 4,096 direcciones por cada pie cuadrado del planeta tierra.

La proliferación de los palmtops, teléfonos celulares, la necesidad de aplicaciones QoS y la disponibilidad del Internet para negocios, y hasta los electrodomésticos que necesitan de direcciones IP es lo que han llevado y contribuido a esta necesidad de que el IPv4 evolucione al IPv6.

Mientras todavía hoy día el enfoque es en computadoras para la necesidad de direcciones IP en el futuro, más y más dispositivos serán conectados a las redes y al Internet. Esto llevará a una explosión en el número de direcciones requeridas por la población, esto sucederá aunque la población de los usuarios del Internet vía navegación no creciese más, lo cual es imposible.

### Repaso del IPv6

Para poder responder la necesidad de más direcciones IP, el RFC 1883 expandió el IP a la versión 6 (IPv6). El IPv6 empieza ofreciendo enrutamiento expandido y capacidad de direccionamiento mediante el uso de un espacio de dirección de 128-bit comparado con al espacio actual de direcciones de 32-bit. Esto permite simplificar la autoconfiguración de direcciones y a LANs usar una dirección MAC como la porción de host de una dirección para eliminar conflictos.

El IPv6 permite unicast, anycast y multicast pero no broadcast. Esto reducirá el flujo de tráfico y la sobrecargas en el sistema y los procesadores de los routers. Además le ofrece un simple formato cabecera en el cual las tareas de cheksum son pasadas a las capas más elevadas del modelo OSI así para que los enrutadores y los switches se dediquen ha hacer sólo su trabajo.

El IPv6 ofrece soporte para opciones, capacidades de Calidad de Servicio (QoS), autentificación y capacidades de privacidad.

Para optimizar el flujo a través de los enrutadores, IPv6 ha reducido el número de campos en los cabezales del IP y ha introducido el concepto de extensiones de cabezales para manejar las funcionalidades menos usadas y para separar la información necesitada en los enrutadores de la que necesita en el host destinatario.

Los tamaños de los campos también han sido tomados en consideración para permitir alineamiento sobre fronteras naturales, lo cual facilita para que una parte del procesamiento se lleve acabo en el hardware.

El IPv6 no tiene checksum de cabezales; el cuenta con los protocolos encapsulados para llevar esto acabo, lo que significa que los enrutadores a lo largo de la ruta no tienen que implicarse con llevar acabo las calculaciones de checksum.

Las direcciones ahora son de 128-bits y pueden ser divididas en tres principales grupos por funciones:

| | |
|---|---|
| ***Unicast*** | ***Direcciona un único host*** |
| ***Multicast*** | ***Direcciona múltiples hosts simultáneamente*** |
| ***Anycast*** | ***Direcciona por lo menos uno de un número de hosts*** |

Los enrutadores que detectan que un datagrama anycast ha sido recibido no necesita remitir el datagrama. No hay broadcast ya que esta funcionalidad pueda ser proveída con multicasting a todos los hosts en una LAN.

**Formato del Cabezal IPv6**

El formato del cabezal del IPv6 es diferente del formato actual del cabezal del IPv4, como es ilustrado en la gráfica a continuación.

**FOTOOOOOOOOOOOSSOOSOSOSOOSOOS**

Los primeros bits tienen un significado predefinido:

| | |
|---|---|
| Version | 4-bit Protocolo de Internet número de versión igual a 6 |
| Priority | 4-bit Valor de Prioridad |
| Flow Label | 24-bit Campo |
| Payload Length | 16-bit Entero sin signo dando la longitud de la carga (Payload Length), ej: el resto del paquete después del cabezal IPv6 en octetos. |
| Next Header | 8-bit Selector; identifica el tipo de cabezal inmediatamente después del cabezal IPv6; usa los mismos valores como el campo de protocolo IPv4. |
| Hop Limit | 8-bit; Entero sin signo; disminuido por uno por cada nodo que reenvía el paquete; el paquete es descartado si el limite de salto (Hop Limit) se reduce a cero. |
| Source Address | 128 bits; la dirección de quien envía inicialmente el paquete. |
| Destination Address | 128 bits; la dirección de quien ha de recibir el paquete finalmente (posiblemente no el ultimo receptor si un enrutamiento opcional es presente) |

La etiqueta de flujo (Flow Label) aún es experimental pero puede ser usada para identificar todos los

paquetes asociados con un flujo de información o conversación. Los enrutadores pueden hacer uso de esto para reducir el procesamiento de data para el flujo. Esto puede permitir decisiones hechas para el primer paquete en el flujo a ser mapiado rápidamente para toda la data subsecuente en el flujo.

Las direcciones IPv6 son escritas en hexadecimal con dos puntos separando cada dos octetos, de este modo, hay 8 piezas de 16-bits.

## Asignación de Dirección

Con el el cambio del tiempo las asignaciones de IP han ido cambiando. El nuevo esquema de Asignación ocurrira en la version 6, como se muestra acontinuacion.

La actual Asignación de dirección IPv6 es como se muestra en la tabla.

| Asignación | Prefix (Binario) | Fraccion |
|---|---|---|
| **Reservado** | **00000000** | **1/256** |
| **Direcciones de red ISO** | **0000001** | **1/128** |
| **Direcciones de red Novell (IPX)** | **0000010** | **1/IZB** |
| **Provedores basados en dirección Unicast** | **010** | **1/8** |
| **Direcciones Unicast EOT reservadas basadas en geografía** | **100** | **1/8** |
| **Direcciones de uso de vinculo local** | **1111111010** | **1/1,024** |
| **Direcciones de Uso de Sitio local** | **million** | **1/1,024** |
| **Direcciones Multicast** | **nn mi** | **1/256** |

Nota que más del 70% aun esta siendo asignada!

### Extensión de Cabezales

Las opciones será colocadas en un cabezal de extensión separadas. La mayor parte de los cabezales no serán procesados hasta que el paquete no llegue a su final de la destinación. Esto lleva acabo un incremento de velocidad por encima de la versión 4. Además, debido a la longitud arbitraria de la extensión del cabezal, futuras características adicionales podrán ser soportadas a medida que se conviertan necesarias.

Esta construcción permite más de lo siguiente:

- Enrutamiento Extendido (igual que IPv4 loose-source route)
- Fragmentación y reensamblamiento
- Seguridad de integridad y autentificación
- Confidencialidad
- Opciones especiales que requieren procesamiento de salto a salto (hop-by-hop)
- Información opcional a ser examinada por el nodo destinatario

Mejora para el soporte para extensiones y opciones incluyen lo siguiente:

- Cabezales Opcionales
- Flow Labeling- Método aún experimental de mantener opciones de punta-a-punta para conexiones mediante todos los enrutadores participantes:
    - Fragmentación sólo por el host de origen- Mínima MTU de 576 bytes en todos los vínculos
    - Debe soportar entrega de vínculo MTU.
    - Uso recomendado de PATH MTU Discovery (RFC 1981).
- Administración de ancho de banda- Campo prioritario de valor de 8 a 15 para data de tiempo real.
- Capacidades de autentificación y privacidad (RPCs 1826 y 1827)

### ICMPv6

En ICMP, al igual como en otras partes de la versión 6, también experimentará cambios:

- Mensajes de error de ICMPv6:
  1 Destino no alcanzable
  2 Paquete demasiado grande
  3 Tiempo excedido
  4 Problema de parámetro

- Mensajes de información de ICMPv6:
  128 Responder petición
  129 Responder respuesta
  130 Consulta de Membrecías de Grupo
  131 Reporte de Membrecías Grupo
  132 Reducción de Membrecías Grupo

**Integración de la Capas Físicas**

Además hay un componente de integración de la Capa Física en el IPv6, ajustada en el RFC 1970. La Capa física será integrada lógicamente con la capa de Internet. Los siguientes servicios y puertos son cambiados:

- El Descubridor de Router es remplazado y mejorado por los puertos 133 (Solicitud de Router) y el 134 (Anunciador de Router).
- arpa es remplazado por los puertos 135 (Solicitud de Vecino) y el 136 (Anunciador de Vecino).
- Redireccionamiento de ICMP es colocado por el puerto 137 (Redirect).
- Un multicast especial formado es usado en lugar del broadcast arpa.

**La Transición al IPv6**

La transición al IPv6 no puede ser realizada con un simple script de implementación o en una rápida y única fase. La transición a IPv6 tiene que ser necesariamente lenta y metódica porque el Internet es demasiada amplia para poder efectuar estos tipos de cambios bruscos.

Debido a el inmenso número de servidores, enrutadores, redes, IPSs, ISPs, etc., que son parte del Internet, todos los usuarios tienen que habilitados para poder migrar individualmente en su propio tiempo y su propia forma. Es por esta razón que tiene que haber interoperatividad entre los dos el IPv6 y el IPv4, el estándar actual.

**Ambientes Mixtos de IPv4 e IPv6**

Dos formas especiales de direcciones son necesarias durante el período de transición:

- IPv4 Compatible, usadas en sistemas que puedan soportar IPv6.
  0000;0000:0000:0000:0000:0000:0102:0304 puede ser escrito ::0102:0304 o ::1.2.3.4
- IPv4 Mapiado, usado por sistemas que no soportan IPv6.
  0000;0000:0000:0000:0000:FFFF:0102:0304 puede ser escrito ::FFFF:0102:0304 o ::FFFF:1.23.4

Los formatos dados permiten protocolos de transporte puedan generar checksums correctamente en direcciones mapiadas. Si el que envía genera un checksum IPv4, el cual es más adelante mapiado, el recibidor sería capaz de verificarlo como un checksun IPv6 mapiado.

**IPv4 Cliente, IPv6 Servidor**

Durante el período de transición, habrán reglas especiales aplicados a todas las conversaciones en el Internet. En los casos donde un cliente IPv4 intente contactar un servidor IPv6, se aplicaran las siguientes reglas de contacto:

- El servidor es contactado por el cliente con una dirección IPv4. Los servidores deben tener dos direcciones mapiadas (una versión 4 y una 6) para poder responder las peticiones de los clientes desde ambos tipos de máquinas. El stack IPv4 mapea las direcciones, pero las capas superiores ven IPv6.
- Todos los sockets en el servidor son IPv6.
- Las reglas de contacto para las máquinas clientes que soportan IPv6 son mucho más simple; ¡Los clientes IPv6 simplemente usan IPv6!
- Sólo se espera del cliente IPv4 conozca IPv4 y buscará una dirección IPv4 para sus servidores.

Esta Búsqueda puede ser buena vía DNS- el cliente haría una petición de un record A donde los clientes IPv6 hicieron peticiones de records AAAA. Para que esto trabajo, el servidor debe tener una dirección IPv4 asignada; sin embargo, mientras que este esta preguntando internamente, una de las direcciones intranet será usada.

**IPv6 Cliente, IPv4 Servidor**

En situaciones donde el cliente soporta IPv6 y el servidor no, se aplican los siguientes reglas:

- El cliente contacta el servidor en una dirección IPv4 mapeada a una dirección IPv6.
- La Búsqueda de nombre de host retornaría direcciones mapiadas a hosts IPv6 demandando records AAAA para servidores IPv4.
- El stack de IPv4 mapea las direcciones.
- La capa superior ve IPv6.
- Todos los socket del cliente son IPv6.

Estos procesos son ilustrados en el siguiente diagrama.

**(99999999999999999**

Los clientes IPv4 simplemente usan IPv4 como ellos han hecho todo el tiempo.

El cliente IPv6 demanda una dirección AAAA para los servidores y el DNS contesta con una dirección IPv4 mapiada; esto causa el stack IPv4 a ser usados en comunicación con el servidor.

***Tunneling Automático***

Una de las más robustas características del IPv6 y una de las más interesantes, es el tunneling automático. Con el tunneling, por si no recuerda, es el proceso de establecer un canal seguro entre los nodos de Internet.

Los border routers en las fronteras de los sistemas IPv6 puede encapsular la data IPv6 dentro de un paquete IPv4 (estableciendo el campo del protocolo a 41). Si las direcciones usadas son compatibles con IPv4, entonces el paquete IPv4 sería enviado a la dirección IPv4. Esto es llamado tunneling automático.

***Tunneling Configurado***

IPv6 además soporta tunneling configurado. En contraste al tunneling automático, los border routers en cada extremo deberán saber información explicita de configuración. Si direcciones IPv4 compatibles no están disponibles, entonces los boundary routers necesitarían información de configuración explicita para suplir el gateway que los paquetes encapsulados son entregados, el enrutador destinatario. Esto es llamado tunneling configurado es simplemente pasar un tunnel IPv4 entre dos redes ejecutando IPv6.

**Ejercicio 2-9: El IPv6 y el IPv4**

No se proveen soluciones a este ejercicio.

Una aplicación ejecutándose en un cliente (A) implementando IPv6 envía data a un servidor (B) que implementa IPv4.

1.- Describa como la data avanza a través de la pila TCP/IP en ambos A y B y que dirección de destino es usada por B en cada etapa (no se preocupe por la dirección IP de A para este problema).

## Ejercicio 2-10: Concerniente al IPv6

No se proveen soluciones a este ejercicio.

Es necesario proveer un reporte de estrategia de redes, y en particular un problema de asignación de dirección y como su COMPAÑÍA puede continuar proveyendo nuevas direcciones IP para los nuevos hosts durante la creciente expansión.

1.- La solución es obviamente migrar hacia el IPv6, ¿pero como?
Pruebe a ver si puede sugerir una estrategía que cubra los siguientes puntos:

A.- ¿En cuantas etapas por separado cambiaría usted por completo la red de la COMPAÑÍA?

B.- ¿En que orden cambiaría usted cada una de esta etapas?

C.- ¿Donde es lo más probable que necesitaría usted nuevo equipo/software?

D.- ¿Dejaría usted redes bajo el IPv4 en cualquier parte de la red de la COMPAÑÍA?

E.- ¿Cual otro beneficios derivados del IPv6 podría usted ofertar a los diferentes departamentos además de la no falta de direcciones IP?

F.- ¿Puede usted también tomar ventaja de las características del IPV6 para ofertar algunas mejoras potenciales en el futuro inmediato?

## RESUMEN

En este capítulo, discutimos los siguientes conceptos básicos de redes:

- Como interactúan los diversos dispositivos de LAN/WAN, incluyendo los NICs, repetidores, hubs, bridges, routers, brouters, switches, gateways, CSU/DSU, modems, y los patch panels
- Las diferencias entre medios comunes de transmisión usados en redes, como son los cables par-trenzados, coaxial, y fibras ópticas y el medio inalámbrico (wireless)
- Los tipos de transmisión, incluyendo asincrónico y sincrónico, simplex, half duplex, full duplex, baseband, y broadband
- Las diferencias entre la topologías lógicas y física
- Los Estándares LAN, incluyendo los del IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), Apple LocalTalk y FDDI (Fiber Distributed Data Interfase).
- Los conceptos fundamentales del direccionamiento IPv4 de 32-bit, incluyendo la estructura de la direc-

ción IP, clases de direcciones, reglas de direccionamiento y direcciones reservadas.
- Convertir direcciones binarias a formato decimal y al contrario.
- Crear mascaras de subnet para acomodar a las necesidades de una compañía y determinar los rangos de las direcciones IP de la subred.
- Las diferencias entre el IPv4 y el IPv6
- Estándares WAN, incluyendo el X.25, frame Relay, y Asynchronous Transfer Mode (ATM)
- Las funciones y los tipos de sistemas carriers T-carrier y E-carrier

## PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué archivo controla la configuración del inetd?
2. ¿Cuáles son las tres clases de direcciones IP?
3. ¿Qué hace el Protocolo de Resolución de Direcciones(arpa)?
4. ¿Qué son los tres componentes para la conexión del PPP?

# CAPITULO 3

*REDES DEL SISTEMA OPERATIVO GNU/LINUX*

# ENRUTAR/ROUTING


| TÓPICOS PRINCIPALES | No. |
|---|---|
| Objetivos | 68 |
| Preguntas Pre-Examen | 68 |
| Introducción | 69 |
| La Redes a Simple Vista | 70 |
| Subnetear/Subnetting | 81 |
| Conceptos de Enrutamiento | 89 |
| Protocolos de Enrutamiento Dinámico | 92 |
| IP Multicast | 110 |
| Resumen | 190 |
| Pregunta Post-Examen | 191 |


## Objetivos

Al finalizar este capítulo, usted estará preparado para efectuar las siguientes tareas:

- Estar familiarizado con las tareas de configuración- opciones de enrutamiento, redireccionamiento de IP (forwarding) y configuración del kernel- asociado con usar a GNU/Linux como un router.
- Describir los rol de ARP y las tablas de enrutamiento en la configuración de interfases de redes.
- Describir la instalación, el propósito y la implementación de firewalls.
- Describir la instalación e implementación de un servidor proxy.
- Describir el rol de broadcasting, asignación de direcciones y multicast en la configuración de interfases de redes.
- Describa los pasos involucrados en IP multicast así como las situaciones donde este método es útil.

## PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué es una netmask (mascara de red)?
2. ¿Cómo es que funciona enrutamiento/routing?
3. ¿Por qué es que usamos enrutamiento Dinámico?
4. Dentro de una red Clase C, ¿Cómo podemos direccionar más de 255 máquinas?
5. ¿Cuándo es que un gateway se convierte en un firewall?

# INTRODUCCIÓN

En este mundo interconectado por el Internet, el concepto de routing y packet forwarding juegan un papel importantísimo. No me puedo imaginar como se pudieran conectar todas estas computadoras y redes en el gran gigante que es el Internet sin la ayuda de los routers. Cada paquete que es emitido por cada computadora individual debe ser direccionado de alguna manera u otra para que pueda llegar a su destino. Cuando se da el caso y esta destinación no es dentro de la red LAN local, los routers se tienen que involucrar. En este capítulo, discutiremos como es que esto es llevado acabo y como es que esto funciona. Otra idea muy importante relacionada con enrutamiento es subnetting o subnetear. Ella permite que manejemos más efectivamente un grupo existente de direcciones IP y para facilitar conectividad entre segmento de la misma red pero construidas sobre diferente topologías. También cubriremos en una manera más breve la idea opuesta de subnetear que es supernetear/supernetting, la cual, por lo general, es para aliviar la carga sobre los routers del backbone, reduciendo el tamaño de la tablas de enrutamiento. Echaremos un vistazo al amplio tópico de filtrado de paquetes más específicamente al concepto de servidores de firewalls y proxy. Finalmente, repasaremos los conceptos más importante de enrutamiento Dinámico y los tipos de protocolos de enrutamiento.

## REDES A SIMPLE VISTA

En los días de desarrollo de las redes nos encontrábamos con sistemas UNiX a larga distancias interconectados sobre líneas telefónicas estándar con varias instalaciones de líneas dedicadas para ellos. Estos sistemas usaban el protocolo TCP/IP para facilitar la comunicación. Con la integración de más y más computadoras en redes, TCP/IP se convirtió y continua siendo, el protocolo por defecto del Internet. Cuando nos referimos a “Internet” y “Internet” son dos cosas diferentes. El Internet es el nombre correcto de lo que se refiere como la conglomeración mundial de computadoras y redes. Una Internet se refiere a dos redes conectadas juntas.

Las siglas TCP/IP significan Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo de Internet). Cada parte del TCP/IP maneja un aspecto diferente del transporte de la data. El TCP maneja la diagramación (layout) de la data que se envía a través de la red asegurandose que el layout de la data enviada es recibida apropiadamente. El protocolo IP maneja el transporte de la data a su destino. Una analogía común es la de un documento enviado por el correo tradicional. El TCP representa la carta dentro del sobre, mientras que el IP representa la dirección escrita en el sobre.

IP es efectivo porque enruta los paquetes al computador receptor y a la red específica. Todo esto se logra usando una dirección IP de 32-bit en el cabezal del paquete. Interpretar un número de 32-bit no es un tarea fácil; la dirección es comúnmente anotada usando un formato de decimal separado por punto, Ej., 192.168.2.34. Pero claro esta, los humanos recuerdan nombres más fácil que recordar números. Debido a esto, existen librerías de funciones que proveen información de los hosts para nombres o números. Cual parte de la dirección denota un red o un nodo es determinado por la mascara. Por ejemplo, una dirección IP de 192.168.33.99 xon un mascara (netmask) de 255.255.0.0 denota una red de 192.168 y 33.99 es la computadora o host en particular en la red. Dependiendo de las necesidades del sitio, más direcciones pueden ser necesitadas; así pues que, el sitio puede implementar subnetting. Con subnetting, una dirección representa una red. Un computador asignan las direcciones IP internas a las computadoras en la red. Todas las peticiones son pasadas a través de un único host en la red interna al Internet y vice versa.

En esta próxima sección, discutiremos herramientas de red.

### Herramientas de Red

GNU/Linux incluye varios utilitarios para diagnosticar, configurar y monitorear el estado de la red. Para más información, deberá consultar las páginas man de los comandos. Cubriremos cada uno de las siguientes herramientas más adelante en este libro:

- **ifconfig**
  Este es el comando usado para configurar el dispositivo de interfase de red. Con este comando, podemos asignar direcciones TCP/IP además de activar y desactivar el dispositivo. Si escribimos "ifconfig" sin ninguna opciones nos muestra la configuración de los dispositivos activos. Agregando la opción -a despliega el estado de todos los dispositivos, activos o inactivos.
- **netstat**
  Este comando muestra información sobre el estado de la red incluyendo las tablas de enrutamiento, conexiones y estadísticas de las interfases. Escrita en la línea de comando sin ninguna opción, nos despliega todos los sockets activos.
- **ping**
  El utilitario ping envía paquetes de data a un host en el sistema. Este comando ayuda en la determinación de la disponibilidad de un host en la red.
- **route**
  Sin opciones, el comando route nos muestra las tablas de enrutamiento. Este comando es usado para establecer direcciones TCP/IP estadística a otras máquinas.
- **traceroute**
  Este comando nos devuelve la ruta que un paquete toma a través del sistema y gateways en su camino al computador de destino.

### SUBNETTING/SUBNETEAR

En redes TCP/IP, existe la necesidad de dividir redes en secciones. Las razones para segmentar un red incluyen proximidad de recursos, balanceado de carga de tráfico en la red, organización unidades de sistemas autónomos, o por otras variadas razones. Por estas razones y muchas un poco otras que están más allá del alcance de este libro, redes están en la necesidad de crear estas divisiones. En el ámbito de las redes TCP/IP, esto se lleva acabo con la implementación de subnetting.

Los siguientes tópicos son discutidos en esta sección:

- Conectar Redes
- Direcciones IP
- Definir Subredes/Subnetworks
- Ejemplos de Subnet
- Implementar una Subred
- Supernetting/Supernetear
- Classless Interdomain Routing/Enrutamiento de Interdominios sin Clase (CIDR)
- Consecuencias de las Tablas de Enrutamiento

### Conectar Redes

Las redes pueden ser subdivididas en subredes lógicas. Desde el punto de vista del IP, los componentes de una dirección IP describen una única red. Esto no siempre es conveniente. Por ejemplo, considere una organización con varias LANs pequeñas, las cuales deben ser unidas para dar acceso al Internet.

En este caso, es posible subdividir una red de IP único en un número de subredes más pequeñas, llamadas subnetworks, donde cada una de estas, comparten la misma dirección IP de red. Para el mundo fuera de la red, sólo existe una red y toda la administración de enrutamiento, etc., puede ser manejada desde el nivel local.

Usando este mecanismo, también es posible subdividir los hosts en una red física en grupos lógicos que pueden ser considerados subredes. Este método es menos común pero puede también ser muy útil.

**Direcciones IP**

Una dirección IP es un número binario de 32-bit. Esto implica que se puede direccionar hasta 4 billones de hosts. Históricamente las redes Clase A, B, y C fueron libremente asignadas, y sólo una fracción de las direcciones en cada red son usadas. Esto resulto en un escaces de direcciones IP para las nuevas organizaciones. Unas solución de corto a mediano plazo fué el uso de subredes.

En la infancia del Internet nadie envisionó que esta creciese a los niveles que ha adquirido. Consecuentemente, compañías como IBM y la Hewlett-Packard que estuvieron en el desarrollo del Internet desde sus comienzos fueron otorgadas redes de Clase A completas.

La gran mayoría de estas organizaciones poseen un exceso de estas direcciones que fueron puestas a su disposición. Una sola red Clase B completa puede proveer 65,000 direcciones de IP para sus hosts. Muy poca organizaciones tienen m´s de unos cuantos miles de hosts en una red, así que un infimo porcentaje de las direcciones son usadas actualmente.

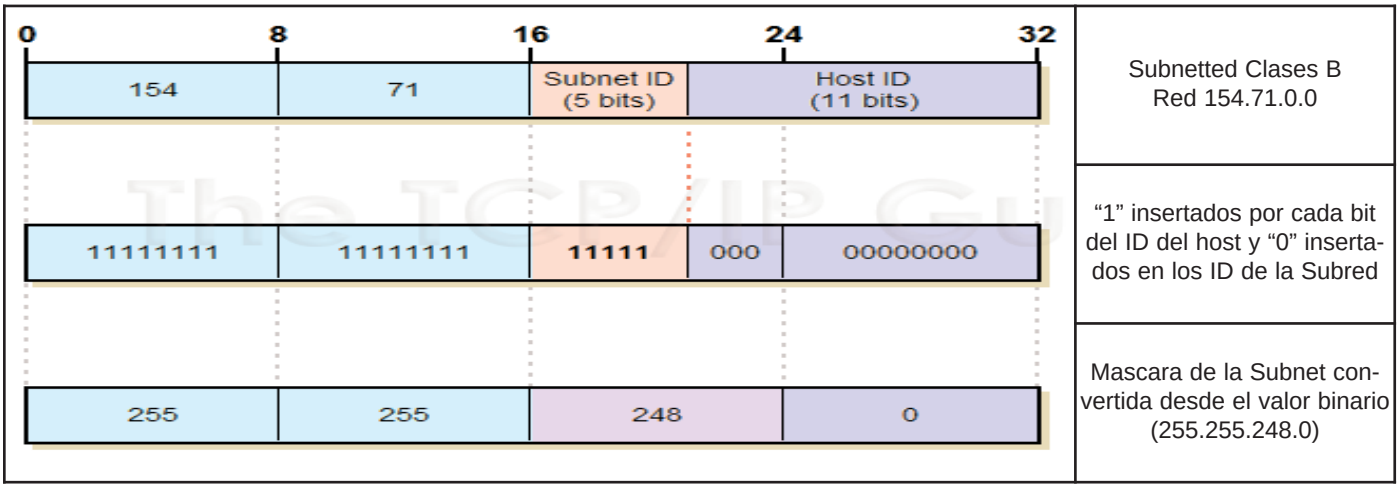
Como se mencionó anteriormente, una solución disponible es incrementar el número de bits en las direcciones IP a más de 32 así definiendo una nueva versión de IP, llamado el IPv6. Aunque esto no ayuda a las implementaciones de las ya existentes redes de IPv4.

**Definir Subredes/Subnetworks**

Es una tarea muy simple establecer subredes en su red d área local. Es llevado acabo con el uso de las mascaras de red/netmask. La netmask es una cadena de cuatro octetos de unos y ceros que es usada para enmascarar el componente de dirección de red en una dirección IP, así separandola de la porción host de la dirección. Esta cadena es especificada utilizando una anotación de decimales separados por punto. Por ejemplo:

**255.255.0.0**

Esta es la mascara por defecto de una dirección Clase B, donde los dos primeros octetos constituyen la dirección de red.




Con la manipulación de la mascara de red, es posible alterar la porción de la dirección IP que es usada para especificar un número de red. Es muy común, por ejemplo, usar el tercer octeto de una dirección de Clase B como un número de subred. Esto es bien fácil de establecer; simplemente se establece la mascara así:

**255.255.255.0**

Ahora es posible especificar hasta 254 subredes, cada una puede contener hasta 254 hosts y pueden operar independientemente. Esto es por lo general preferible a tener una red única con 65,534 hosts. Al mundo externo la organización aparenta ser una red singular con su número de una red Clase B.

En una situación donde no hace falta subnetear, entonces no se especifican subredes. Use la dirección IP de Clase B exactamente como es definida.

| | |
|---|---|
| **135.33.156.2** | **Dirección IP Clase B** |
| **135.33** | **Dirección de Red** |
| **156.2** | **Número de Host** |

En el siguiente ejemplo, usamos el tercer octeto completo para la dirección de subnet y el cuarto octeto para la porción de la dirección del host. No se necesita efectuar ningún tipo de recalculación.

- Use el primer octeto del número de host como el número de subred.
- Esto nos da 254 subredes con 254 hosts.
- Especifique una mascara de 255.255.255.0.

| | |
|---|---|
| **135.33.156.2** | **Dirección IP Clase B** |
| **135.33** | **Dirección de Red** |
| **156** | **Número de Subred** |
| **2** | **Número de Host** |

Aún cuando subneteamos, algunas direcciones de red mantienen su significado.

• El caso normal

| | |
|---|---|
| **135.33.156.2** | **Dirección IP Clase B** |
| **255.255.0.0** | **Clase B mascara** |
| **135.33** | **Dirección de Red** |
| **156.2** | **Número de Host** |
| **255.255** | **Broadcast dirigido** |

• Subnet case

| | |
|---|---|
| **135.33.156.2** | **Dirección IP Clase B** |
| **255.255.255.0** | **Clase C mascara** |
| **135.33** | **Dirección de Red** |
| **156** | **Número de Subred** |
| **255** | **Broadcast dirigido a la Subred 156** |
| **255.255** | **Broadcast dirigido a todas las Subredes** |

No podemos usar subredes cuando los números de las subredes tiene todos sus bits establecidos a uno. La razón para estos es que la dirección de broadcast en esta subred fuera la misma que la de broadcast para todas las subredes; así que la dirección de subnet con todos los unos es ilegal.

| | |
|---|---|
| **135.33.255.2** | **Dirección IP Clase B** |
| **255.255.255.0** | **Mascara de Subred** |
| **135.33** | **Dirección de Red** |
| **255** | **Número de Subred** |
| **255** | **Broadcast dirigido a la Subred 255** |
| **255.255** | **Broadcast dirigido a todas las Subredes** |

Un argumento similar nos detiene de usar la subnet con el número 0 (el 0 denota la ruta por defecto).

En general, se pierden dos números de subred (todos los unos y todos los ceros) cuando usamos subredes. Una excepción a esta regla es cuando usamos Classless Interdomain Routing (CIDR). Más adelante en este mismo capítulo discutiremos CIDR.

### Ejemplo de Subred

Tomemos la siguiente dirección Clase B:

**137.64.0.0**

La mascara por defecto es 255.255.0.0. Esto nos devuelve 65,534 direcciones de host (descontamos la de todos los unos y la de todos los ceros).

La queremos dividir en por lo menos seis redes.

| # of bits | Subnet Mask | # of Subnets | # of Hosts/Subnet |
|---|---|---|---|
| 1 | 255.255.128.0 | 2(0) | 32,766 |
| 2 | 255.255.192.0 | 4(21 | 16.382 |
| 3 | 255.255.224.0 | 8(6) | 8,190 |

Una mascara de subred de 255.255.224.0 nos da ocho redes, de las cuales sólo seis son útiles.

| | | | |
|---|---|---|---|
| 137.64.0.0 | 137.64.32.0 | 137.64.64.0 | 137.64.96.0 |
| 137.64.128.0 | 137.64.160.0 | 137.64.192.0 | 137.64.224.0 |

Examinemos un caso más complejo de subnetear. Nuestros requerimiento es dividir una red única lógica en por lo menos seis redes (Ej., administración, ventas, nomina, soporte, producción y recursos humano). La mejor manera de hacer esto es encaminarse a incrementar la mascara del de la subred hasta que descubra cual mascara mejor cumple con sus requisitos.

La primera mascara a tratar es la de 1-bit, 255.255.128.0, la cual nos da dos posible redes, 137.64.0.0 y 137.64.128.0, de las cuales ninguna son válidas. Esto claro esta no satisface nuestro requerimiento, así que debemos extender la mascara.

| | | | | | |
|---|---|---|---|---|---|
| Netmask | 255.255.128.0 | 11111111 | 11111111 | 10000000 | 00000000 |
| Network 0 | 137.64.0.0 | 10001001 | 01000000 | 00000000 | 00000000 |
| Network 1 | 137.64.128.0 | 10001001 | 01000000 | 10000000 | 00000000 |

La segunda mascara a tratar es una de 2-bit, 255.255.192.0. Esta mascara nos da cuatro posible redes, dos de las cuales no son válidas (todos cero o todos uno), lo cual aún no llena nuestro requisito.

| | | | | | |
|---|---|---|---|---|---|
| Netmask | 255.255.192.0 | 11111111 | 11111111 | 11000000 | 00000000 |
| Network 0 | 137.64.0.0 | 10001001 | 01000000 | 00000000 | 00000000 |
| Network 1 | 137.64.64.0 | 10001001 | 01000000 | 01000000 | 00000000 |
| Network 2 | 137.64.128.0 | 10001001 | 01000000 | 10000000 | 00000000 |
| Network 3 | 137.64.192.0 | 10001001 | 01000000 | 11000000 | 00000000 |

Ahora probamos con una con mascara de 3-bit, 255.255.224.0. Esta nos da ocho posible redes, dos de las cuales son inválidas, lo cual si satisface nuestros requerimiento y nos sobran dos!

| | | | | | |
|---|---|---|---|---|---|
| Netmask | 255.255.224.0 | 11111111 | 11111111 | 11100000 | 00000000 |
| Network 0 | 137.64.0.0 | 10001001 | 01000000 | 00000000 | 00000000 |
| Network 1 | 137.64.32.0 | 10001001 | 01000000 | 00100000 | 00000000 |
| Network 1 | 137.64.64.0 | 10001001 | 01000000 | 01000000 | 00000000 |
| Network 3 | 137.64.96.0 | 10001001 | 01000000 | 01100000 | 00000000 |
| Network 4 | 137.64.128.0 | 10001001 | 01000000 | 10000000 | 00000000 |
| Network 5 | 137.64.160.0 | 10001001 | 01000000 | 10100000 | 00000000 |
| Network 6 | l37.64.192.0 | 10001001 | 01000000 | 11000000 | 00000000 |
| Network 7 | 137.64.224.0 | 10001001 | 01000000 | 11100000 | 00000000 |

***Ejemplo de Subnetear en una Clase C***

La red Clase C 197.20.4.0 es mostrada aquí, subneteada en ocho subredes (se muestran sólo cuatro).

**Subnet Mask = 255.255.255.224**

Aquí podemos ver una red Clase C subneteada en 8(6) subredes. Donde sólo cuatro están siendo usadas en el diagrama en la imagen, lo que nos deja dos redes en caso de crecimiento o expansión.

Las direcciones de red tal vez le parezcan extrañas ya que no terminan en un cero. Esto es causa de que

la porción host de la dirección ahora solo es de 6 bits en longitud. Los dos bits de arriba del ultimo byte en la dirección ahora establecen el número de la subred, esto nos da las combinaciones 0, 32, 64, 96, 128, 160, 192, y 224, de los cuales 32, 64, 96, 128, 160, y 192 son subredes validas que podemos usar.

Ahora podemos subnetear cuanto sea necesario, mientras tenga suficiente direcciones de host en la subredes que estén en uso.

**Implementando una Subred**

Implementar una subred es un proceso que toma dos pasos. El primero, establecer la mascara de la subnet en su intefase. La mascara de subnet es usada para todas las rutas vía el interfase. Use el comando ifconfig o una herramienta gráfica equivalente para establecer la mascara del subnet. El segundo es, agregar sus entradas a la tabla de enrutamiento. Debe crear una entrada por separado para cada dirección de subnet usando el comando route o una herramienta gráfica equivalente.

| Destination | Gateway | Flags | Refcnt | Use | Interfase |
|---|---|---|---|---|---|
| 137.64.0 | 137.64.32.21 | UG | 2 | 8,989 | eth0 |
| 137.64.32 | 137.64.32.3 | U | 12 | 256,448 | eth0 |
| 137.64.64 | 137.64.32.21 | UG | 0 | 487 | eth0 |
| 137.64.96 | 137.64.32.3 | U | 1 | 9,027 | eth0 |
| 137.64.128 | 137.64.32.254 | UG | 1 | 942 | eth0 |
| .... | | | | | |
| 192.9.200 | 137.64.32.254 | UG | 1 | 96,196 | eth0 |

En nuestro ejemplo, el host esta en la red 137.64, y hemos decidido que requerimos un mascara de subnet de 255.255.224.0. Si nuestra dirección de host es 137.64.32.3, esto significaría que estamos en el tercer host 0.3 en la subred 137.64.32. Para implementar una subnet, deberá imponerle la mascara de subnet que ha calculado a la interfase que maneja los paquetes de esa red. Debemos usar el comando apropiado para alterar la mascara de subnet asociada a la intefase. Tome en cuenta que la mascara de la subnet puede ser que no se le presente cuando despliegue la información del enrutamiento. El uso de la mascara es implícito debido a la ruta que esta siendo asociada con la interfase.

Una vez ha efectuado el cambio de la mascara de la subnet, usted podrá agregar rutas para cada subnet. Generalmente, estas rutas apuntarán a uno o más enrutadores en su subred local. Pero, usted puede si desea, tener varias subredes en el segmento que esta directamente conectado, aunque esto si trabaja en contra de los beneficios de subnetear.

Por lo general, todos los hosts en una red usaran la misma mascara de subnet. Ciertos softwares, como el RIP, pueden sentirse confuso si este no es el caso.

**Super-netear/Super-netting**

Donde Subnetear es la adición de más bits de redes desde nuestra disponibilidad de bits de hosts, supernetear es simplemente el inverso. Si VLSMs (Variable Length Subnet Masks) son soportadas, podemos entones acortar la mascaras así como alargarlas (RFCs 1517 hasta 1519). Ahora acortamos el número de bits asignado a la red.

Las tablas de enrutamiento pueden ser reducidas agrupando rutas con los prefijos similares debajo de una sola entrada en la tabla. La longitud del prefijo puede ser seleccionada para hacer así el grupo tan grande como sea posible. Todas las direcciones de destino com el mismo prefijo pueden compartir una misma ruta. Es útil principalmente no para configurar redes pero para usar en las tablas de enrutamiento para reemplazar bancos consecutivos de direcciones de redes con una sola dirección de red de un nivel más alto. Esto acorta la tabla de enrutamiento enormemente. En vez de una mascara de subnet para enmascarar la parte de red de

la dirección, usamos la llamada supermascara (supernet mask).

**CIDR- Enrutamiento de Interdominio sin Clase (Classless Interdomain Routing)**

CIDR hace uso de supernetting (RFCs 1517 hasta 1519). CIDR esta soportada por la versión 4 del Protocolo Border Gateway (BGP4) y es usada para reducir el tamaño de las tablas de enrutamiento en los routers del backbone. CIDR fué propuesto en el 1993, ya cuando las direcciones de redes habían sido repartidas con una medida de disponibilidad alrededor de todo el mundo. Rutas más específicas (con mascaras de red más largas) pueden ser usadas para sobrescribir el enrutamiento para redes asignadas en áreas geográficas incorrectas por el mé explicado anterior de disponibilidad en vez de áreas geográficas. El CIDR requiere que las direcciones de redes asignadas en el futuro sean en base a sus área geográfica. La IANA (The Internet Assigned Numbers Authority) distribuye bloques de direcciones a centros regionales. El centro regional entrega bloques de direcciones IP a los ISPs. La entidad ARIN (American Registry for Internet Numbers) administra a los Estados Unidos y, aunque su nombre insinúa diferente, a los otros paises (esta área de asignación de números antes era manejada por el InterNIC).

## Consecuencias de las Tablas de Enrutamiento

Asumamos que un ISP desea asignar dieciséis redes Clase C a un cliente. Bajo condiciones normal esta situación requeriría dieciséis rutas. Si las redes tienen números consecutivos, sólo una ruta es requerida al usar supernetting.

**FOTO FOTO FOTO=====**

Aquí podemos apreciar como el uso de supernetear en CIDR es una gran ayuda para los routers backbone. Como este método puede drásticamente ayudar a reducir el tamaño de la tabla de enrutamiento. Pero si vale la pena mencionar que depende en la disponibilidad de bloques de direcciones consecutivas.

Esfuerzos a grande escala están siendo llevados acabo para reorganizar la colocación de direcciones de IP. Una de las ideas básicas es la asignación de conjuntos de direcciones Clase B a diferentes países y así simplificar la tarea de enrutar entre naciones.

**Ejercicio 3-1: Subnetting/Subnetear**

En este ejercicio, no necesitamos estaciones de trabajo. No se proveen soluciones a este ejercicio.

Tomemos las siguiente información en consideración antes de subnetear en ambiente de una compañía X. Esta compañía a decidido expandir y está en la necesidad de crear nuevos departamentos que necesitan ser interconectados a la red.

En la actualidad hay dos departamentos usando la siguiente configuración:

- Depto 1 usa 201.40.25.0 red de Clase C y tiene suficiente direcciones para hasta 20 hosts.
- Depto 2 usa201.40.26.0 red de Clase C que tiene suficiente direcciones para hasta 35 hosts.

Deseamos agregar dos Departamento más: uno que tiene 15 hosts y el otro con 35 hosts.

El único problema es que no tenemos más redes Clase C disponible para usarlas en estos nuevos departamentos. Tendremos que emplear una solución de subneteo para tomar las Clase C ya existentes y dividirlas para acomodar los nuevos hosts.

Además, debemos planificar cambiar para sólo usar una red Clase C única ya que en el futuro necesitaremos más redes para otros departamentos.

Deberá planificarse para:

- El cambio al nuevo esquema de subredes en cuatro departmento usando dos redes Clase C.

• El cambio a cuatro departamento con una red única Clase C.

Detalle cual departamento usará cual red, la mascara de la subnet a usar y cuantos hosts tenemos disponibles y con que rango de direcciones IP. Tome en cuenta cualquier router que desearía usar entre los departamentos para llegar a sus conclusiones.

## CONCEPTOS DE ENRUTAMIENTO

Un entendimiento de los conceptos de enrutamiento es critico para su entendimiento de enrutar como ciencia.

Los siguientes tópicos son discutidos en esta sección:

• Enrutamiento de Red
• Interconectar Redes
• Envío de Paquetes IP /IP Packet Delivery (Red Local)
• Enrutador o Pasarela/Gateway
• Envío en Redes Multiple
• Tabla de Enrutamiento IP
• Orden de Búsqueda en la Tabla de Enrutamiento
• Desplegar la Tabla de Enrutamiento
• Establecer las Rutas
• Filtrado de Paquetes
• Servidor Proxy de SOCKS
• Comandos route y traceroute
• Modo Promiscuo

### Enrutamiento de Red

¿Qué es enrutar? Los paquetes tienen que ser entregados desde una computadora a otra a través de las redes. El destino puede ser una máquina local o una remota. En cualquier de los casos, los paquetes tienen que ser enrutados desde la máquina fuente hacia la máquina destino.

***Igual que el departamento de clasificación de la Oficina de Correo***

Enrutar es el corazón de las redes modernas de TCP/IP. Podemos compararlo con el modelo de enviar una carta a través del sistema postal. Si deseamos enviar una carta, se nos ofrecen en esencia dos maneras:

• Si la carta es para alguien en su misma calle, usted puede caminar y colocarla en su buzón personalmente.
• Si está fuera de su calle, la coloca en el buzón de salida de la calle.

Las misma dos opciones se presentan a un computador en una LAN. Esta puede o entregar el paquete directamente a las máquinas en la red local o entregarla al enrutador que sabe como obtener la información acerca de la dirección de destino.

**Interconectar Redes**

Recuerde que todos los hosts en una red deberán tener una dirección IP única. La dirección contiene los componentes:

• La parte Red
• La parte Host

El componente de la dirección que es red será igual para todos los hosts en una red en particular. así que, en este diagrama tos los hosts en la red superior tienen una dirección IP con el componente de red de 192.9.100. Similarmente todos los hosts en la red inferior tienen dirección IP con el componente de red

de192.9.200.

Para poderse comunicar entre host A (192.9.100.2) y host B (192.9.200.4), deberá existir una manera de transferir los paquetes IP desde la red192.9.100 a la red 192.9.200 y vice versa.

**Envío de Paquetes IP (Red Local)**

En el siguiente ejemplo:

- Usted esta en el punto A.
- Usted mismo puede entregar una carta al punto B, ya que B le queda en la misma calle.
- Usted entonces no usa el buzón de salida.

Entregarle correspondencia a alguien en su misma calle es una tarea simple, y como es costumbre no involucra el correo oficial. Este es el mismo proceso que ocurre cuando se transfiere data entre dos computadoras en la misma red local.

La tarea de reenviar un paquete hacia su destino final es manejado por el IP sin el conocimiento de los protocolos de más alto nivel de transporte.

En un caso simplificado, donde el paquete esta siendo enviado a un sistema en la misma red, el IP simplemente pasa el paquete a la capa de Enlace de Data, dirigiendola hacia su destino. En este caso, las tramas de enlace de data (data-link) y los datagramas de IP son direccionados a mismo sistema de destino. En el ejemplo anterior, A le da el paquete IP a la interfase de red. Entonces, la interfase de red coloca el paquete IP en una trama Ethernet direccionada a B. Luego, la trama Ethernet es enviada en la red local. Aquí B recoge la trama Ethernet con su dirección de destino y extrae el paquete IP, entonces el paquete IP se envía para arriba hacia la capa de Transporte.

**Enrutador o Pasarela**

La terminología varia, pero en realidad no existe la gran diferencia, entre un enrutador (router) y una pasarela (gateway). Históricamente, un gateway se refiere a una computadora con capacidades de enrutamiento habilitadas, mientras que un router es un dispositivo dedicado físico. Pero gateway/router conectan dos o más redes. así que, este tiene más de una interfase de red y respectivamente más de una dirección IP. El gateway/router debe ser visible en ambas redes y cada dirección IP esta asociada con la interfase de la red. Los host que cumplen esta condición a menudo también son referidos como hosts multihomed.

**Network A:192.9.100 Host-1: 192.9.100.99 Network B:192.9.200 • Host-2:192.9.200.99**

En primera instancia estas dos redes deberán estar conectadas físicamente. Esto se logrará usando un equipo que tendrá dos interfases y estará conectado a ambas redes. Tal sistema es conocido como un gateway.

Además de estar físicamente conectado a las dos redes, el gateway debe poder ser direccionado desde ambas redes. En pocas palabras, Este debe parecer como un sistema normal de la red 192.9.100 y también de la red 192.9.200. Para esto ser posible, el gateway deberá tener dos direcciones IP, una que es válida en la red 192.9.100 y la otra que sea válida en la red 192.9.200.

Las direcciones IP son asociadas con las interfases de redes apropiadamente conectadas a la red y esto significa que el gateway puede comunicarse con ambas redes.

**Envío en Multiple Redes**

En el siguiente ejemplo:

- Usted entrega una carta a su oficina de correo en el buzón local, porque B esta en una calle diferente a la suya.
- El servicio postal la recoge y la lleva oficina de clasificación.
- Entonces es entregada a B.

En este ejemplo no estamos enviando correo a alguien que vive en nuestra misma calle. El problema desde nuestro punto de vista, no es más complejo. Nosotros simplemente ponemos la carta en el buzón local y es todo.

El servicio postal llega, recoge la carta desde el buzón local, lo toma para la oficina de clasificación. Aquí la dirección escrita en la carta es usada para decidir donde se debe enviarla. Esta puede ser enviada a otra oficina para aún más clasificación, dependiendo que tan lejos esta deberá viajar.

Al la carta llegará a su destino B entregada por el servicio postal.

Este mismo proceso también es el que es aplicado a los paquetes de una red de computadoras para la entrega de paquetes.

Fijese que la dirección de la capa de Internet no ha cambiado- la carta aún es dirigida a B- pero el emisor primero tiene que entregarle la carta a un tercero,la oficina de correo.

Si el destino se encuentra en una red diferente a la de la fuente, el IP debe reenviar el paquete hacia el sistema que actúa como un gateway en esa red. Para hacer esto, el datagrama de IP es encapsulado en una trama de enlace de data con destino al equipo gateway. Cuando el equipo recibe el datagrama, se da cuenta que realmente no esta destinado para el propiamente y entonces procede a usar su propia tabla de enrutamiento para ver si lo puede reenviar a su destino. Esto puede involucrar enviarlo directamente a su destinatario en la misma red que el proxy o quizás enviarlo más adelante a otro gateway si aún se encuentra en una etapa intermedia.

```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2       192.168.2.7     255.255.255.0   U     0      0        0 eth0
200.42.200.0    200.42.100.1    255.255.255.0   U     0      0        0 eth1
196.3.81        192.168.2.7     255.255.255.0   U     0      0        0 lo
```

En este ejemplo, el paquete necesita viajar por el router (C):

1.- A sabe que B no esta en la Red 1, así que envía el paquete IP en una trama Ethernet direccionado a C, a la dirección por defecto para los paquetes dirigidos a los hosts, A no sabe esto ya que esto es función interna de C.

2.- C extrae el paquete IP desde la trama Ethernet, lo inspecciona, y descubre que su destino final no es C.

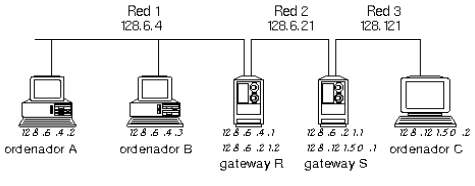
3.- C es local a ambas A y B, y por esto puede colocar el paquete IP en otra trama Ethernet en la Red 2 y la direcciona a B. Note como C tiene dos direcciones de diferente Ethernet en este caso- una en la Red 1 y otra en la red 2.

4.- B recibe la trama de Ethernet que proviene de C, le extrae el paquete IP, y lo envía hacia arriba a la capa de Transporte.

**Tabla de Enrutamiento IP**

EL IP toma su decisión en enrutar paquetes usando una estructura de data llamada tabla de enrutamiento (routing table). El formato de su tabla de enrutamiento depende de cual sistema usted tiene en uso.

La tabla de enrutamiento contiene las entradas para cualquier red que el sistema local tiene conocimiento. Este da la dirección del sistema de gateways para cada una de las redes que desea conectarse. El gateway debe estar físicamente conectado a la misma red como el host local. La entrada también específica cual interfase de red deberá ser usada para enviar los paquetes al gateway.




La primera línea muestra una entrada para la red local. En este caso, el gateway esta en el sistema local. En la segunda línea nos muestra la ruta desde la 200.42.100 a la red 200.42.200. En este caso, el gateway esta físicamente conectado a la red de destino. La tercera línea contiene la ruta a una red que no tiene el gateway conectado directo a ella, o sea a ambas 192.168.2 y 196.3.81. En este caso, la tabla de enrutamiento contiene la dirección del gateway que lo lleva a la próxima etapa de la ruta.

**Un Ejemplo de Enrutamiento**

El computador A puede enviar datagramas al B directamente, usando la red 1. Sin embargo, no puede llegar al ordenador C directamente, puesto que no están en la misma red. Hay varias maneras de conectar redes. En el gráfico asumimos el uso de gateways (existen otras alternativas). En este caso, los datagramas que van desde A a C deben ser enviados a través del gateway R, red 2 y gateway S. Todos los ordenadores que usan TCP/IP necesitan que se les suministre la información y algoritmos apropiados para que puedan saber cuándo un datagrama debe ser enviado a través de un gateway, y elegir el gateway apropiado.

**Tabla de enrutamiento de A:**

A es local a B y R por esto puede entregar el paquete directamente a B. Pero no es local a C por eso lo envía vía R.

**Tabla de enrutamiento de B:**

B es local a ambos A y R, por eso entrega paquetes directamente a ambos. Para llegar a A, esta lo entre directo.

**Tabla de enrutamiento de C:**

C es local al host S. así que, este no puede entregar paquetes a ningún host directamente que no sea S.

| La tabla de enrutamiento para un host 21.30.20.0 con router 21.30.20.0 a otras redes | Destino | Ruta | Tipo de Ruta |
|---|---|---|---|
| | 21.30.20.0 | Entrega Directa | Network |
| | 136.125.1.0 | Via 21.30.20.1 | Network |
| | 198.224.3.1 | Via 21.30.20.1 | Host |
| | 198.224.3.2 | Via 21.30.20.1 | Host |
| | 201.12.30.0 | Via 21.30.20.1 | Network |

**Tabla de enrutamiento de R:**

R es local a A y B y a S y por eso puede entregar paquetes directamente a ellos. Este no es local a C, así que este puede enviar paquetes para C vía S.

**Tabla de enrutamiento de S:**

S es local a C y a R y por eso puede entregar paquetes directamente a ellos. Este no es local a A ni a B, así que este puede enviar paquetes para A y B vía R.

| La tabla de enrutamiento para un host 21.30.20.0 con router 21.30.20.0 a otras redes | Destino | Ruta | Tipo de Ruta |
|---|---|---|---|
| | 21.30.20.0<br>0.0.0.0 | Entrega Directa<br>Via 21.30.20.1 | Network<br>Default |

**Rutas a Redes**

Para grandes redes, especificar la dirección IP de cada host en la tabla de enrutamiento se convirtiera en inmanejable debido a su tamaño. Imaginese, que, cada dirección IP en el Internet tenga que ser incluido en una tabla de enrutamiento. La única solución es usar rutas a las redes. Estas son simple direcciones de red en una tabla de enrutamiento.

No es nuestro interés tener que especificar una ruta a todas las diferentes redes en el mundo. Si existe un paquete con una dirección de destino que la tabla de enrutamiento no contiene una ruta especificada, usaría la ruta por defecto. Es por lo normal la última entrada en la tabla de enrutamiento. La dirección IP 0.0.0.0 se refiere a la ruta por defecto en la tabla de enrutamiento.

| | |
|---|---|
| Rutas Hosts | Mascara más larga |
| Rutas Redes | Mascara más Corta |
| Ruta por Defecto | |

La rutas por defecto agregan otro nivel de simplificación a tablas de enrutamiento. Si no existiese tal cosa como la ruta por defecto, necesitaríamos especificar la ruta a todas las redes que desearíamos contactar en cualquier momento- imaginese todas las redes en Internet.

## Orden de Búsqueda en Tabla de Enrutamiento

El orden en el cual la tabla de enrutamiento es buscada es de extrema importancia. La búsqueda siempre se efectúa desde la más específica hacia la menos específica.

Esto es controlado por la mascara para la ruta en la tabla de enrutamiento. Mascaras más largas, esto quiere decir más específicas, son igualadas con preferencias comparadas con las más cortas o menos específicas.

| Destination | Route | Type | Metric |
|---|---|---|---|
| 127.0.0.1 | 127.0.0.1 | Host | 0 |
| 21.30.20.0 | 21.30.20.10 | Network | 1 |
| 21.30.20.0 | 21.30.20.20 | Network | 2 |
| 0.0.0.0 | 21.30.20.1 | Default | 5 |

Por ejemplo la segunda ruta con el metric más pequeño (1) será usada ya que es la más rápida.
La ruta con el metric 2 será usada sólo si el hosts 21.30.20.10 no está disponible

Aprenderemos más sobre las más caras más adelante, pero por ahora, mantenga presente que al final es la mascara de la ruta que es usada para determinar la igualdad en la tabla de enrutamiento, no se asume nada acerca de la clase o tipo de rutas involucradas.

### Metrics

Metric es una medida de cuantos gateways/enrutadores (hops/saltos) un determinado paquete tendrá que atravesar en su viaje para arribar a su destino. Esta medida es usada para elegir una ruta donde existe más de una opción disponible. Esta ayuda para determinar cual ruta es la más corta para llegar al destino de para entreg el paquete.

Un metric no tiene que realmente reflejar el verdadero número de saltos involucrado en una ruta. Es simplemente un parámetro que puede ser tan arbitrario como sea necesario.

Es muy útil diseñar que que se pueda elegir entre diferentes posibilidades de enrutamiento, quiero decir, donde proveemos más de una ruta para entregar un paquete dado. Al configurar multiple rutas a un mismo destino, los paquetes pueden ser entregados aunque una de las rutas este temporalmente fuera de servicios por cualquier razón.

Fijese en en esta tabla anterior, la ruta por defecto ha sido arbitrariamente establecida con un metric de valor 5; aunque este host esta conectado al Internet, el número real de saltos que un paquete tendrá que efectuar para llegar a su destino pueden ser hasta 30.

### Desplegar la Tabla de Enrutamiento

Para desplegar la tabla de enrutamiento desde la línea de comando, podemos usar el comando route o el comando netstat con la opción -r.

```
# route
kernel   IP routing Table
Destination     Gateway         Genmask         Flags   Metric  Ref    Use   Iface
255.255.255.255 *               255.255.255.255 UH      0       0      0     eth0
192.168.1.0     *               255.255.255.0   U       0       0      0     eth0
127.0.0.0       *               255.0.0.0       U       0       0      0     lo
default         tricom.asdl     0.0.0.0         UG      1       0      0     eth0
# netstat  -r
kernel   IP routing Table
Destination     Gateway         Genmask         Flags   MSS     Window  irtt  Iface
255.255.255.255 *               255.255.255.255 UH      0       0       0     eth0
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **192.168.1.0** | ***** | **255.255.255.0** | **U** | **0** | **0** | **0** | **eth0** |
| **127.0.0.0** | ***** | **255.0.0.0** | **U** | **0** | **0** | **0** | **lo** |
| **default** | **tricom.asdl** | **0.0.0.0** | **UG** | **1** | **0** | **0** | **eth0** |

En GNU/Linux (y algunos otros sistemas operativos), usted puede usar el comando netstat para ver la tabla de enrutamiento. Los detalles de esto se cubren en los ejercicios asociados con este capítulo. Muchos de los comando de redes son iguales o muy similares a los de todas las distribuciones de GNU/Linux o los diferentes sabores de UNiX.

## Estableciendo las Rutas

La tabla de enrutamiento pueden ser mantenidas manualmente por el administrador de sistemas usando el comando route. Este comando es mayormente usado para inicializar un sistema con la ruta a gateway por defecto en la red. El sintaxis de este comandos es:

**route comando destino [netmask] [gateway] [metric] [dispositivo]**

En este sentencia donde comando es uno de dos agregar o eliminar (add o delete) destino es la dirección IP a entregar el paquete, netmask es la mascara de la ruta el gateway es la dirección de la pasarela, metric es para elegir entre diferentes gateways con el mismo destino y dispositivo es para elegir entre un dispositivo y otro. Aquí le presentamos unos cuantos ejemplos que deben aclarar esto un poco:

```
route add -net 192.166.1.0 netmask 255.255.255.0 eth1
route add -net 10.10.0.0 netmask 255.255.255.0 eth2
route add default eth0
```

### *Enrutar es un Camino de dos Vias*

Uno de los problemas más comunes de enrutamiento al configurar las tablas de enrutamiento es entender que debe ser efectuado en cada punto. Un escenario es Máquina A trata de hacerle ping a Máquina B. A ha sido configurada correctamente para enviar los paquetes de B a través del gateway R. R también ha sido configurada para enviar los paquetes entre las dos redes y entregar tanto a A como a B. Pero, B no no tiene en su tabla de enrutamiento como entregar los paquetes a A. así pues que no sabe como responder la petición del comando ping entrante de A. así que A no recibirá ninguna respuesta de su ping o cualquier otro servicio de red porque N no sabe como enviarle los paquetes de respuesta.

La solución es decirle a B que envíe los paquetes de A vía el router R. Luego los paquetes podrán fluir apropiadamente en ambas direcciones y el ping funcionará.

## Filtrado de Paquete

Filtrado de paquete controla cuales paquetes IP son permitidos arribar a ciertas direcciones. Puede ser usado para limitar acceso foráneo a las computadoras locales y vice versa. En GNU/Linux, el filtrado de paquetes se controla a nivel del kernel. Existen módulos para el kernel que permiten definir un sistema de reglas para aceptar o rechazar los paquetes o las comunicaciones que pasan por el sistema. Estos sistemas de reglas conforman lo que se conoce como firewall o cortafuegos; en otros sistemas los firewall pueden estar implementados en software y estar desvinculados del sistema operativo, pero en el caso de linux, el firewall se puede montar a nivel de kernel y no es necesario instalar un software adicional que para más a veces tiene agujeros.

La gran mayoría de los servicios de Internet son almacenar y enviar (como es el correo y las noticias) o interactivos (como el FTP, WWW):

- El firewall normalmente será un host de reenvío para los servicios de almacenar-enviar. por ejemplo, el

correo se le envía al firewall, el cual lo reenvía hacia la red cual el protege.
- Son los servicios interactivos que requieren del uso explicito de proxies.

***Las ipchains***

En versiónes de Kernel anteriores a la 2.4 se disponía de los módulos IPCHAINS para montar firewalls, así como de otros módulos de soporte para comunicaciones concretas (enmascaramiento de FTP, de IRC, Real Audio etc.).

Gracias a IPCHAINS, además de poder definir normas de firewall muy precisas, también podía hacer que un linux funcionara como gateway enmascarando todas las peticiones de una LAN.

El filtrado de paquetes es proveído al nivel del Kernel en GNU/Linux por la herramienta ipchains (versión del kernel 2.2 y anteriores) y iptables (2.4 en adelante), ipchains funciona revisando reglas para determinar que hacer con determinado paquete. Tres chains son internas (built in): input, output, y forward. La cadena de entrada (input chain) es revisada cuando un paquete arriba. Si un paquete va hacia otra máquina, la cadena de reenvío (forward chain) es revisada. Justamente antes de que un paquete salga, la cadena de salida (output chain) es revisada.

Manejabilidad, los usuarios pueden definir sus propias cadenas y agruparlas con cadenas ya existentes. Por ejemplo, si el firewall se conecta al computador local vía el dispositivo eth0 y al Internet vía el dispositivo ppp0, uno puede definir las cadenas ppp-out, ppp-in eth-out y eth-in y agruparlas con las cadenas correspondientes.

Para definir una nueva cadena use el comando ipchains -N nombre-cadena. Las reglas pueden ser insertadas o agregadas en una cadena con las opciones -I -A.

**Firewalls**

Las técnicas de conta-fuego o firewalls son las más comunes para proveer seguridad de redes. El tópico de conta-fuego es muy amplio, así que aquí solo cubriremos brevemente los temas involucrados.

La manera más simple de un firewall es un router con el reenvío de paquetes no disponible. La meta principal es proveer protección del mundo externo. En algunos casos el administrador de sistemas puede decidir restringir el acceso a los usuarios al mundo externo. Como resultado, el firewall incrementa el rendimiento de los servicios y el tráfico al limitar otros. Una desventaja obvia de los firewall es la manera en que ellas restringen tráfico que se dirige fuera de la red al igual que en el que entra.

Una máquina conectada a dos redes es llamada un multihome host (esto quiere decir que tiene homes en dos redes y puede ver ambas). Un firewall es simplemente un host multihomed que tiene la opción de reenvía de paquetes apagada. Con el reenvía de paquetes desabilitado, el firewall puede ver ambas redes puro los host en esas redes no pueden ver esos host en el otro lado del firewall. En vez de desabilitar todo el reenvío de paquetes algunos firewall desabilitan casi todos los paquetes pero permiten que algunos útiles pasen. Este firewall esta ejecutando funciones de enrutamiento y es conocida como un screening (filtrado) router. El firewall es diseñado para proteger una red del acceso a usuarios no deseados del exterior. Podemos verlo como un par de mecanismo: uno que bloquea tráfico y otro que lo permite. El énfasis en uno o el otro se determina por la política de seguridad.

El problema de poder asegurar cada hosts en una red es simplemente demasiado dificultoso y consume demasiado tiempo, claro dependiendo del tamaño de la red. Aunque es simple tarea asegurar quizás unos seis hosts en una red pequeña, no es el caso asegurar una red quizás con unos 2,200 hosts, esto simplemente fuese una pesadilla, especialmente que se estuviese que enfrentar la individualidad de cada host.

Es mucho más fácil y simple asegurar una red completa, que asegurar los clientes individuales. Para lograr esta situación, conectamos la red al mundo externo vía un firewall, el cual bloquea toda la comunicación entre las redes. Los hosts en cada red pueden ver sólo el firewall, pero no se pueden ver entre ellos. Aún debemos asegurar un host, claro esta el firewall- ya que este es visible al mudo fuera de nuestra red.

**Servidores Proxy**

Los firewalls son fantásticos en cuestión de seguridad, pero tienen una pequeña falla, y es que la red interna tampoco puede ver fuera de ellos! así que necesitamos poder selectivamente transmitir algunos paquetes IP a través del firewall. Un Servidor Proxy es exactamente lo que necesitamos. Ellos re-transmiten paquetes enviados fuera de la red pero solamente ciertos tipos de paquetes (e.j., WWW).

Los servidores proxys nos permiten comunicarnos a través de un firewall enviando solamente un pequeño conjunto de nuestros mensajes. Ellos permiten solamente un conjunto de transacciones, prohibiendo todo el otro tráfico. El beneficio de un proxy es que solamente permite que tome lugar un tipo de comunicación, por ejemplo peticiones FTP o WWW. así como canalizando todas las peticiones a través de una pequeña ventana el proxy permite que cada petición sea registrada en un diario a un más, por lo general los proxys son configurado para solo permitir peticione salientes, nunca entrantes. Fijece que los servidores proxys no incrementan la seguridad; sino que la reducen. Hay un mecanismo de proxy genérico llamado SOCKS (que significa sockets, lo cual discutiremos más adelante) que permite crear proxies para cualquier servicio de Internet.

***Los Proxies Tradicionales***

Con los proxies tradicionales, los paquetes que provienen del Internet nunca entran directamente la red local y los paquetes de la red local nunca entran directamente al Internet. La conexión al Internet solo se puede efectuar a través del firewall. Cualquier servicio que uno desea que sea accesible desde el Internet (como es un servidor Web) debe estar en el firewall. Una aplicación en la red local debe estar informado de la existencia de un proxy y conectarse explícitamente al proxy en el firewall. Esto puede ser problemático ya que algunos software no soportan el uso de servidores proxies y por esto, no funcionarían correctamente, si es que funciona. SOCKS es un ejemplo de un servidor proxy tradicional.

***Proxies Transparentes***

Los proxies transparentes no requieren que las aplicaciones sepan de su existencia. El proxy debe aún estar ejecutándose en el firewall, pero las aplicaciones simplemente se conectan como siempre y los paquetes son transparentemente redirigidos al proxy cuando pasan por el firewall. El Squid es un ejemplo de un servidor proxy que puede ser configurado para ser un servidor proxy transparente.

Podemos por ejemplo redireccionar conexiones salientes desde la red local hacia el proxy en el firewall usando ipchains. Por ejemplo:

```
# ipchains  -A input  -p tcp -s 192.1.2.0/24 -d / BO -j  REDIRECT 8080
```

La dirección IP de origen 192.1.2.0/24 es usada en este ejemplo. El 24 le dice a ipchains que solo los primeros 24 bits son importante ya que la red completa esta siendo referenciada y no un solo host. Esto redireccionará todo el tráfico desde la red local hacia el puerto 8080 en el firewall, donde un servidor proxy estará esperando.

**Enmascarando (Masquerading) el IP**

Los paquetes de la red local traviesan todo el Internet de punta a punta, pero sólo por tratamiento especial que reciben. Software de proxy no es necesaria si usamos enmascarado de IP, pero una utilidad

especial del kernel llamada enmascarando es usada. Los paquetes desde la red local en su ruta hacia el Internet son rescritos por el firewall para que así ellos parezcan haber originado desde el firewall. Los paquetes desde el Internet hacia la red local son también rescrito para que así ellos aparezcan haber haber originado en el servidor proxy y simplemente estén dirigidos hacia el computador cliente.

Mucho de los programas clientes no entiende a los proxies, así que no pueden ser usados. Esto incluyen muchos clientes FTP. Este problema se resuelve usando enmascaramiento de IP. La configuración de un proxy muchas veces depende de un navegador Web como es mozilla o Netscape para todos los requerimientos ya que ellos si entienden los proxies.

En el siguiente ejemplo creamos una cadena ppp-in, la soldamos a otra cadena input y establecemos las políticas por defecto del dispositivo ppp0 al valor de DENY, lo cual significa que solo específicas conexiones las cuales fueron habilitadas posteriormente en la configuración son permitidas:

```
# ipchains -N ppp-in
# ipchains -A ppp-in -j DENY
# ipchains -A input -j ppp0 -1 ppp0 -j ppp-in
```

En el siguiente ejemplo permitimos conexiones ser hechas al servidor web llamado http: //www.portal.com:

```
# ipcliains -A ppp-in -p TCP -d www.portal.com 80 -j ACCEPT
```

Antes de que enmascara del IP funcione, el reenvío de IP (IP forwarding) debe ser establecido ejecutando el comando echo 1>/proc/sys/net /ipv4/ip_forward. La dirección IP es asignada dinámicamente, como es PPP o SLIP, el comando echo 1 >/proc/sys/net/ipv4/ip-_forward debe ser ejecutado. Luego, asegurandose que la interfase externa se llame ppp0, ejecute el siguiente comando:

```
# ipchains -P forward DENY
# ipchains -A forward -i ppp0 -j MAS0
```

Estos pasos establecerán una configuración básica de enmascaramiento de IP.

## Las iptables

IPCHAINS ha muerto, viva IPTABLES. A partir del kernel 2.4 se esta dando soporte a otro módulo para filtrado de paquetes mucho más potente que IPCHAINS, llamado IPTABLES. Para acceder a ciertos sites ftp tendremos problemas usando IPCHAINS con el kernel 2.4. A pesar de que IPCHAINS siga funcionando, ya no tendremos los antiguos módulos para solventar los problemas de acceso a servicios especiales y se recomienda pasarse a IPTABLES.

### *Diferencias respecto a IPCHAINS*

IPTABLES es más completo que IPCHAINS, permite un control aún más preciso, aunque también es más complejo.

En principio el sistema sigue siendo el mismo. Hay que cargar un módulo del kernel (se puede hacer en el propio script de firewall), y ejecutar un script de shell convencional que tiene el aspecto de un conjunto de reglas. Un script de este tipo se podría complicar y sofisticar tanto como se deseara, eso ya es cuestión de cada cual.

El script generalmente: · Comienza cargando los módulos necesarios (los imprescindibles y los auxiliares, como el de ftp masquerading), · Establece algún bit como por ejemplo el de forwarding. · Luego borra todas las reglas actuales (flush). · Establece las políticas por defecto para la aceptación, reenvío y salida. · Y finalmente va aplicando todas las reglas de firewall, que varían enormemente dependiendo de las necesidades de cada red. El orden de algunos puntos no tiene por que ser siempre así.. Por lo general, una aproximación buena suele ser CERRAR todo por defecto, e ir abriendo lo que se necesite. Aunque esta última aproxi-

mación da mucho más quebraderos de cabeza.

***Diferencias respecto a IPCHAINS.***

- La sintaxis, obviamente, aunque no mucho.
- DENY no existe, ahora sería DROP.
- MASQ y REDIRECT no existen como destinos de paquetes.
- REJECT extendidos con más opciones
- LOG con más opciones, muy útil para monitorear y depurar
- ... y más que se pueden ver en el Howto y en otras páginas.

***Elementos básicos***

***-Ordenes básicas:***

**iptables -F : efectivamente, flush de reglas**
**iptables -L : si, listado de reglas que se están aplicando**
**iptables -A : append, añadir regla**
**iptables -D : borrar una reglas, etc...**

***-Ejemplo de regla:***

**#Regla que acepta conexiones al puerto 80**
**iptables -A INPUT -i eth0 -s 0.0.0.0/0 -p TCP —dport www -j ACCEPT**

***ANATOMÍA DE LA REGLA:***

**iptables: commando iptables (no hay que olvidar que las reglas son un shell script)**
**-A: append, opción para añadir la regla**
**INPUT: estado del paquete (al entrar es input).**

**-i eth0: interfaz de red eth0**
**-s 0.0.0.0/0: dirección de acceso (cualquiera en este caso)**
**-p TCP: tipo de puerto**
**--dport: puerto de destino**
**-j ACCEPT: destino del paquete (se acepta, podría ser DROP, LOG, REJECT,..)**

**-Guía rápida de flags:**
**-s : source address. Ej: -s 192.168.1.0/24**
**-d : destino. Ej: -d 84.56.73.3**
**-p : tipo de protocolo(TCP,UDP,ICMP). Ej: -p TCP**
**--sport : puerto de origen**

**--dport: puerto de destino**
**-i = -in-interfase : el interfaz por el que se entra (eth0,eth1, ppp0,...)**
**-o = --out-interfase: el interfaz por el que se sale (eth0,eth1, ppp0,...)**
**-Notas:**
**-i se usa con reglas INPUT y FORWARD**
**-o se usa con reglas FORWARD y OUTPUT**

A partir de estas normas básicas, conociendo la anatomía básica de una regla, y viendo ejemplos ya tenemos suficiente material para hacernos con el dominio de IPTABLES.

***Ejemplos de Configuración***

***Ejemplo de Firewall Simple***

Ahí va volcado:

**#!/bin/sh**

```
#########################################################################
## SCRIPT de IPTABLES                                                  ##
## Pello Xabier Altadill Izura I+D+I+I en tiempo record                ##
## Investigación, Desarrollo, Innovación e IMPLANTACIÓN                ##
## Este script es de ejemplo y no es el mejor ejemplo,                 ##
## pero funciona en RedHat 7.2 y es muy pedagógico                     ##
#########################################################################

## Notas para usuarios de IPCHAINS:
 # ipchains e iptables son módulos del kernel que
 # NO pueden convivir juntos
 # DENY ahora es DROP
 # Los LOG se guardan de otra forma
 echo -n Aplicando Reglas de Firewall...

## Instalando módulos
 modprobe ip_tables
 modprobe ip_nat_ftp
 modprobe ip_conntrack_ftp

## Variables
 EXTIF="eth0" # La que va al router
 INTIF="eth1" # La que va a la LAN

## Primeras reglas
 /sbin/iptables -P INPUT ACCEPT # INPUT se acepta por defecto MAL HECHO
 /sbin/iptables -F INPUT
 /sbin/iptables -P OUTPUT ACCEPT # OUTPUT se acepta por defecto, weno..
 /sbin/iptables -F OUTPUT
 /sbin/iptables -P FORWARD ACCEPT # FORWARD se acepta por defecto buf
 /sbin/iptables -F FORWARD
 /sbin/iptables -t nat -F

 ## se deniega 80 y se guarda log (ejemplo)
 /sbin/iptables -A INPUT -i $INTIF -s 0.0.0.0/0 -p TCP —dport www -j LOG —log-prefix "IPTablesFW> "
 /sbin/iptables -A INPUT -i $INTIF -s 0.0.0.0/0 -p TCP —dport www -j DROP

## Acceso al 3128 (proxy squid) desde LAN
 /sbin/iptables -A INPUT -i $INTIF -s 192.168.1.0/24 -p TCP —dport 3128 -j ACCEPT
 # El resto se tira
 /sbin/iptables -A INPUT -i $INTIF -s 0.0.0.0/0 -p TCP —dport 3128 -j DROP

 ## Acceso al 143 desde LAN
 /sbin/iptables -A INPUT -i $INTIF -s 192.168.1.0/24 -p TCP —dport 143 -j ACCEPT

## Acceso al ssh desde la LAN
 /sbin/iptables -A INPUT -i $EXTIF -s 213.195.64.0/24 -p TCP —dport 22 -j ACCEPT

## Acceso al ssh un rango externo
 /sbin/iptables -A INPUT -i $EXTIF -s 213.195.64.0/24 -p TCP —dport 22 -j ACCEPT
 # el resto se tira
 /sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p TCP —dport 22 -j DROP
## Acceso al puerto 25
 /sbin/iptables -A INPUT -i $EXTIF -s 213.191.89.0/24 -p TCP —dport 25 -j ACCEPT
 /sbin/iptables -A INPUT -i $INTIF -s 192.168.1.0/24 -p TCP —dport 25 -j ACCEPT
 /sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p TCP —dport 25 -j DROP
```

```
## FORWARD
 # Que me haga log de todo el forward
 /sbin/iptables -A FORWARD -j LOG

## He aqui el forward para la LAN, una regla mágica
 iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
 # Ese peazo de bit que hay que habilitar
 echo 1 > /proc/sys/net/ipv4/ip_forward
```

*Ejemplo de Firewall más Completo*

Este ejemplo es algo más serio, ya que la regla de input por defecto es DROP. Esta política de reglas es la más segura, ya que por defecto se denegará TODO, y poco a poco se van habilitando las entradas precisas.

```
#!/bin/sh
## SCRIPT de IPTABLES
## Pello Xabier Altadill Izura
echo -n Aplicando Reglas de Firewall...

## Paramos el ipchains y quitamos el módulo
 /etc/rc.d/init.d/firewall stop
 rmmod ipchains

## Instalando módulos
 modprobe ip_tables
 modprobe ip_nat_ftp
 modprobe ip_conntrack_ftp

 ## Variables
 IPTABLES=iptables
 EXTIF="eth1"
 INTIF="eth0"

## En este caso,
 ## la tarjeta eth1 es la que va al ROUTER y la eth0 la de la LAN

## Primeras reglas
 /sbin/iptables -P INPUT DROP
 /sbin/iptables -F INPUT
 /sbin/iptables -P OUTPUT ACCEPT
 /sbin/iptables -F OUTPUT
 /sbin/iptables -P FORWARD ACCEPT
 /sbin/iptables -F FORWARD
 /sbin/iptables -t nat -F

### En principio, si las reglas INPUT por defecto hacen DROP, no haria falta
 ### meter más reglas, pero si temporalmente se pasa a ACCEPT no esta de más.

## Todo lo que viene de cierta IP se deja pasar (administradores remotos...)
 /sbin/iptables -A INPUT -i $EXTIF -s 203.175.34.0/24 -d 0.0.0.0/0 -j ACCEPT

## El localhost se deja
 /sbin/iptables -A INPUT -i lo -j ACCEPT
 /sbin/iptables -A OUTPUT -o lo -j ACCEPT

## Aceptar al exterior al 80 y al 443

# Permitir salida al 80
```

```
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 80 -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 80 -j ACCEPT
# Permitir salida al 443
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 443 -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 443 -j ACCEPT

## SALIDA SMTP - Para que el servidor se pueda conectar a otros MTA
# Permitir salida SMTP
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 25 -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 25 -j ACCEPT

## SALIDA FTP - Para que el servidor se pueda conectar a FTPs
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 21 -m state —state ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 21 -m state —state NEW,ESTABLISHED -j ACCEPT
# ftp activo
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 20 -m state —state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 20 -m state —state ESTABLISHED -j ACCEPT
# ftp pasivo
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 1024:65535 —dport 1024:65535 -m state —state ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —sport 1024:65535 —dport 1024:65535 -m state —state ESTABLISHED,RELATED -j ACCEPT

## El acceso al 19720 desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 19720 -j DROP

## El acceso al 19720 desde dentro, ACEPTADO
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 19720 -j ACCEPT

## El acceso al 19721 desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 19721 -j DROP

## El acceso al SSH desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 1972 -j DROP

## El acceso al SMTP desde dentro, permitido.
/sbin/iptables -A INPUT -i $INTIF -p tcp —dport 25 -j ACCEPT
## El acceso al SMTP desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 25 -j DROP

## Acceso al 80 desde el interior ACCEPTADO PARA DOS IPs
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.11/32 -p tcp —dport 80 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.54/32 -p tcp —dport 80 -j ACCEPT
## Acceso al 80 desde el interior DENEGADO PARA EL RESTO
#/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 80 -j DROP

## Acceso al PROXY
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 8082 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.10.0/24 -p tcp —dport 8082 -j ACCEPT
/sbin/iptables -A INPUT -s 127.0.0.0/8 -p tcp —dport 8082 -j ACCEPT

# Desde el exterior denegado
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 8082 -j DROP

## Acceso a POP3 e IMAP desde el EXTERIOR, DENEGADO
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 110 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 143 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.10.0/24 -p tcp —dport 110 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.10.0/24 -p tcp —dport 143 -j ACCEPT
```

```
## Acceso a POP3 e IMAP desde el EXTERIOR, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 110 -j DROP
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 143 -j DROP
## Acceso al 8082 desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 8082 -j DROP

## FORWARD
# Que me haga log de todo el forward
#/sbin/iptables -A FORWARD -j LOG

# He aqui el forward
## Norma general
##iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.11/32 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.16 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.54/32 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.0/24 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.10.0/24 -j MASQUERADE

# Habilitar el forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
```

***Reglas de Protección***

Tranquilos , aquí no vamos a hablar de la última maza de cartas Magic que me he comprado, ni de formulas para salir airosos jugando a La Llamada de Cthulhu, entre otras cosas porque no juego a esas cosas raras. En esta breve sección listamos algunas reglas para proteger nuestro equipo y por extensión la red, de ciertos ataques muy habituales en las redes como el smurf y otras formas de inundación y DoS.

¿Es recomendable usar todas estas normas? Según como administremos el nivel de paranoia.

Nota: hay que dar valores concretos a las variables $

```
# Deshabilitar broadcast
/bin/echo “1” > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Deshabilitar el ping... quizá discutible.
/bin/echo “1” > /proc/sys/net/ipv4/icmp_echo_ignore_all

# Deshabilitar la redirección del ping
/bin/echo “0” > /proc/sys/net/ipv4/conf/all/accept_redirects

# Registrar los accesos extraños, paquetes falseados, etc..
/bin/echo “1” > /proc/sys/net/ipv4/conf/all/log_martians
# Anti-flooding o inundación de tramas SYN.
iptables -N syn-flood
iptables -A INPUT -i $IFACE -p tcp —syn -j syn-flood
iptables -A syn-flood -m limit —limit 1/s —limit-burst 4 -j RETURN
iptables -A syn-flood -j DROP
# Guardar los accesos con paquetes fragmentados, recurso utilizado para tirar
# servidores y otras maldades (bug en Apache por ejemplo)
iptables -A INPUT -i $IFACE -f -j LOG —log-prefix “Fragmento! “
iptables -A INPUT -i $IFACE -f -j DROP

# Anti-spoofing (falseo de ip origen)
```

```
iptables -A INPUT  -i $IFACE -s $IPADDR -j DROP
iptables -A INPUT  -i $IFACE -s $CLASS_A -j DROP
iptables -A INPUT  -i $IFACE -s $CLASS_B -j DROP
iptables -A INPUT  -i $IFACE -s $CLASS_C -j DROP
iptables -A INPUT -i $IFACE -s $CLASS_D_MULTICAST -j DROP
iptables -A INPUT -i $IFACE -s $CLASS_E_RESERVED_NET -j DROP
iptables -A INPUT  -i $IFACE -d $LOOPBACK -j DROP
iptables -A INPUT -i $IFACE -d $BROADCAST -j DROP
(bien explicado en el script de linuxguruz)
# Proxy Transparent: peticiones al puerto 80 redirigir al SQUID(3128)
iptables -t nat -A PREROUTING -p tcp -s 0.0.0.0/0 —dport 80 -j REDIRECT —to 3128
```

## Servidor de Proxy SOCKS

SOCKS le permite a los usuarios acceso al Internet desde atrás de un firewall. Derivada de la palabra sockets, SOCKS centraliza el control de la red sobre la actividad del Internet, permitiendo acceso limitado a computadoras específicas y a través de autentificación (login) a actividades en particular en el Internet. SOCKS es una herramienta más de administración y protección en una red.

Hay dos partes a los SOCKS: el cliente y el servidor. El programa del lado del cliente se ejecuta en el computador detrás del firewall y se conecta al servidor de SOCKS para hacer particiones de acceso al Internet. El servidor de SOCKS se coloca entre la red detrás del firewall y el Internet. Un cliente contacta el servidor, el cual maneja la petición y transfiere la información entre el cliente y el sitio de Internet.

Cuando un cliente hace una petición para obtener acceso al Internet, este envía la siguiente información:

- Comando de petición de conexión
- Número de Puerto
- Dirección de IP Requerida
- Nombre de Usuario del sistema requiriente

Ya en este punto, la administración de la red entra en una etapa importante. La información recibida desde el cliente es comparada a la de las listas de control, creadas por el administrador. Si el cliente posee privilegios de acceso al Internet, la petición es servida y el servidor de pasa la información al sitio de Internet. La data es enviada de regreso desde el sitio de Internet arriva al servidor SOCKS. El servidor pasa la data al cliente después de revisar los permisos del cliente nuevamente.

SOCKS es gratis en el Internet y puede ser descargado desde el sitio Web http://www.socks.nec.com.

Cada cliente que usa SOCKS debe tener su propio archivo de configuración. Al momento de hacer una petición, el archivo /etc/socks.conf es revisado línea por línea. Una vez la conexión requerida es encontrada, la acción especificada en el archivo es efectuada. El archivo contiene los siguientes comandos:

**deny**

Este específica que la dirección TCP/IP que iguala esta línea debe ser permitida. Pero, usted no debe usar este operador en vez de protección apropiada de firewall. Aunque la dirección está bloqueada de acceso, el cliente puede usar un programa no vinculado a la librería de SOCKS, sobrepasando la seguridad implementada en el servidor SOCKS.

**direct**

Este pasa la dirección TCP/IP especificada a través del firewall sin aplicar ningún filtro.

**sockd**

Este instruye a la librería SOCKS para que la conexión TCP/IP sea enviada al daemon sockd donde, después de revisar, este presumiblemente envía la petición fuera de la red.

El archivo /etc/sockd.conf controla cuales conexiones TCP/IP son pasadas al daemon sockd y eventualmente fuera del firewall. El daemon sockd normalmente se ejecuta en el computador situado entre la red interna y la red externa, por lo general en la máquina que actúa como el firewall.

El archivo /etc/sockd.conf es un archivo de texto plano. Cada línea contiene hasta 1,023 caracteres y utiliza el siguiente formato:

**[allow / deny] [?=auth] [*=username | filename]**
**source-address source-mask**
**[destination-address destination-mask]**
**(.operator destination-port]**
**[:shelI -command]**

**allow/deny**
Esta opción específica si las conexiones en esta línea son permitidas o no.

**?=auth**
La política de identificación especificada al inicio del programa puede ser obviadas con esta opción. Usando ?=I causa que la conexión sea rechazada si el cliente esta usando ident o si el nombre retornado por ident no es el mismo que el que es enviado por el cliente SOCKS. Un ?=i causa que sockd rechace la conexión si el nombre retornado por ident es diferente al nombre enviado por el cliente SOCKS.

***=username | filename**
Esta opción causa que la regla iguale a un nombre de usuario en particular.

**source-address source-mask**
Esta opción permite a igualar una dirección IP única o a un grupo de direcciones IP. la dirección de origen puede ser en la anotación de cuatro octetos separados por puntos (100.99.98.97) o el nombre del host. La mascara de origen debe ser en la anotación separada por puntos.

**operator destination-port**
Un puerto en particular o un rango de puertos TCP/IP puede ser especificado con estas opciones. Por ejemplo, “eq 20” establece una conexión igual al puerto de data del ftp. La siguiente es una lista de los operadores disponibles:

| | |
|---|---|
| eq | Igual a |
| neq | No es Igual a |
| lt | Menor que |
| le | Menor que o Igual a |
| gt | Mayor que |
| ge | Mayor que o Igual a |

**:shell-command**
Esta opción específica cual comando usar cuando la línea es igualada. El comando es ejecutado por el shell bash (/bin/sh).

### route y traceroute

El sistemas GNU/Linux, la configuración de enrutamiento estático se logra a través del uso del comando route. Este permite la modificación o visualización de la tabla de enrutamiento. Esta también puede ser usada para ayudar a diagnosticar problemas de enrutamiento. Otro comando muy útil que puede ser usado para diagnosticar problemas de enrutamiento y conectividad es el comando traceroute.

## Comando route

El comando route de GNU/Linux es por lo general usado para designar una dirección IP estática a una red o computador específica usando una interfase de red. El comando route puede ser usado en conjunto con un sistema que recibe su dirección IP desde un servidor DHCP. El sintaxis del comando route es route [opción] [argumento] [dispositivo]. Aquí le presentamos algunas de las opciones más comunes usadas con el comando route. Para una lista más completa refierase a la páginas man del comando route.

| | |
|---|---|
| **-n** | **Despliega los hosts por dirección IP (no se muestran los nombres de host)** |
| **add** | **Agrega una a la tabla de enrutamiento** |
| **del** | **Elimina una ruta en la tabla de enrutamiento** |
| **net** | **Conecta la ruta a una red** |
| **host** | **Conecta la ruta a un máquina host** |
| **gw** | **Específica por cual gateway enrutar en el host o en el sistema de red** |

Este es un ejemplo de algunas de estas opciones:

**# route add -host 192.168.1.0 eth0**

En este ejemplo conectamos a otro host en la red 192.168.1.0 usando la primera interfase de Ethernet.

Al escribir el comando route sin ningunas opciones nos muestra la configuración de la tabla de enrutamiento actual. Por ejemplo:

**# route**
**The following is the Kernel IP routing table:**

| Destination | Gateway | Genmask | Flags | Ref | Use | Iface |
|---|---|---|---|---|---|---|
| 192.168.1.0 | 0.0.0.0 | 0.0.0.0 | U | 0 | 0 | eth0 |
| default | gw.serv.com | 0 | UG | 0 | 0 | eth0 |

La primera y segunda columnas muestran las direcciones IP o los nombres de de el host/red y gateway, respectivamente. La columna Genmask nos da la mascara de la subred para la ruta. La columna Flags tiene tres opciones: U (la conexión esta activa), H (el destino es un host), y G (el destino es un gateway). La columna Metric muestra la distancia al destino, el cual es usado por algunos daemons enrutadores para calcular la mejor ruta. La columna Ref es usada por UNiX para indicar los números de referencia del router. La columna Use muestra cuantas veces el kernel busco la información de enrutamiento. Por último, la columna Iface muestra cual dispositivo de red es usado para mandar los paquetes a esa dirección o ruta.

### Comando traceroute

El comando traceroute es una herramienta invaluable para determinar donde está ocurriendo un problema en la red, traceroute envía un paquete al destino especificado y reporta todos los saltos que se dan en la ruta. Un salto puede no responder por varias razones, incluyendo exceso de tráfico o que físicamente el sitio Web este abajo.

El traceroute utiliza el campo TTL (Time to Live) en el encabezado del paquete para determinar la información de la tabla de enrutamiento. El campo TTL limita a cuantas computadoras o gateways un paquete puede proceder antes de ser descartado. Cada vez que un enrutador reenvía un paquete, esta disminuye el TTL por 1. Si un paquete llegase a cero, el paquete será descartado (dropped) y el router envía un mensaje ICMP haciendole saber al host que fué descartado.

El comando se inicia enviando un paquete con un TTL de 1. Si el primer gateway o host no es el destino asignado, el gateway disminuye el TTL por 1 y envía una respuesta a traceroute. El traceroute da salida a la dirección de la primera repuesta y crea otro paquete. Al nuevo paquete se le da un TTL de 2 y es enviado. El primer gateway pasa el paquete más adelante, reduciendo el TTL por 1. En la próxima pasarela, el TTL, es ahora 1, a este punto, la pasarela reduce el TTL, a 0 y retorna una respuesta al router, traceroute incrementa el TTL, desplegando la repuesta desde los gateways intermedio hasta que el paquete arrive a su destino final. Por defecto, el número de saltos es 30 pero puede ser cambiado usando la opción -m (max_ttl).

Las siguientes opciones son comunes de tracecoute:

**-f Establece el valor inicial TTL, permitiendole saltar enrutadores cerca del host**
**-m Establece el máximo número de saltos**
**-n No efectúa Búsqueda de nombre; sino, sólo da salida a direcciones numéricas de IP**

El comando tracetoute produce salida en un formato similar al del comando ping:

**# traceroute abiertos.org**
**traceroute to abiertos.org (206.123.67.148), 30 hops max, 38 byte packets**
**1 adsl-200-1.tricom.net (200.42.200.1) 1451.603 ms 210.628 ms 90.021 ms**
**2 Leased-Lines-208-93.tricom.net (200.42.208.93) 9.581 ms 11.010 ms 10.798 ms**
**3 Leased-Lines-208-174.tricom.net (200.42.208.174) 13.945 ms 91.611 ms 11.987 ms**
**4 sl-gw11-orl-4-0-TS3.sprintlink.net (160.81.34.65) 139.749 ms 1440.340 ms 1478.104 ms**
**5 sl-bb20-orl-0-0.sprintlink.net (144.232.2.232) 1423.170 ms 1431.891 ms 1431.974 ms**
**6 sl-bb21-atl-10-3.sprintlink.net (144.232.19.73) 1647.392 ms 1576.853 ms 605.996 ms**

```
 7  sl-gw25-atl-15-0.sprintlink.net (144.232.22.14)  126.988 ms  172.011 ms  114.977 ms
 8  sl-timewarner-9-0.sprintlink.net (144.223.62.62)  61.550 ms  1351.235 ms  1546.623 ms
 9  core-01-ge-1-2-0-1.atln.twtelecom.net (66.192.243.40)  1612.145 ms  1522.711 ms  1445.372 ms
10  core-02-so-4-0-0-0.dlfw.twtelecom.net (66.192.255.24)  1606.380 ms  1453.472 ms  1585.293 ms
11  hagg-02-ge-3-3-0-504.dlfw.twtelecom.net (66.192.253.125)  1466.255 ms  1460.830 ms  1525.948 ms
12  66-195-98-102.gen.twtelecom.net (66.195.98.102)  1565.522 ms  156.115 ms  93.964 ms
13  206.123.64.46 (206.123.64.46)  221.905 ms  1633.271 ms  1568.761 ms
14  abiertos.org (206.123.67.148)  467.253 ms  82.485 ms  209.568 ms
```

La primera parte de información es denotar el número de saltos (hops) para llegar desde el origen al router localizado en la dirección dada como argumento. El número de saltos es seguido por el nombre de del router. Si el nombre del router no esta disponible, la dirección IP es desplegada. Luego es la dirección IP del gateway. Los próximos tres campos muestran el tiempo de respuestas, o el tiempo que el paquete toma para hacer el todo el viaje por la red de ida y vuelta desde su sistema y de vuelta.

### Modo Promiscuo

El Ethernet usa un protocolo llamado CSMA/CD (Carrier Sense Multiple Access/Collision Detection (CSMA/CD). El termino Carrier Sense se refiere al proceso por el cual una estación revisa a ver si otra estación esta usando la línea antes de ella enviar. Si otra estación se encuentra ejecutando antes de que la transferencia de data inicie. Si no se detecta data, la estación empieza a transmitir. Si hay tráfico, la estación espera por el próximo tiempo de línea reposo. Acceso multiple significa que cada estación esta conectada a un único cable o conjunto de cables que forman una única ruta. El termino Collision Detection se refiere a la manera por la cual los temas de colisiones son resueltas. Aunque las estaciones revisan por tráfico antes de empezar a transmitir, las estaciones pueden detectar que no hay tráfico y empiezan a enviar simultáneamente. Una vez ocurre una collisión y cada estación se percata de esto (embedido en la data es un medio para detectar si la data llego a su destino que causa que el destinatario responda en un tiempo en específico), ambas estaciones dejan de transmitir y esperan un lapso de tiempo al azar antes de tratar de retransmitir.

Ethernet esta basada en la arquitectura bus. así que, los cables de Ethernet lleva tráfico de data que esta destinado para multiple computadoras. Cada computador escucha y acepta data direccionada específicamente a ella por su dirección MAC. Las computadoras por lo general ignoran paquetes que no estén direccionados a ellos. Pero, un sistema puede ser configurado para escuchar a toda la data que pasa sobre el cable o la data dirigida a otros sistemas. Un sistema esta en modo promiscuo cuando este lee data dirigida a otros sistemas. Existen razones legitimas para que un sistema sea establecido en el modo promiscuo, como puede ser prueba de cables para detectar deficiencias de conexiones análisis de tráfico (tipo de tráfico, los puertos usados , volumen, entre otros.).

### Ejercicio 3-2: Conceptos de Enrutamiento

En este ejercicio, trabajaremos con una tabla de enrutamiento para determinar como los paquetes serán enrutados entre las redes. No se proveen soluciones a es ejercicio.

1. Considere la siguientes entradas en la tabla de enrutamiento.

| Route | Netmask | Destination |
|---|---|---|
| 132.10.56.4 | 255.255.255.255 | 142.12.201.2 |
| 132.10.0.0 | 255.255.0.0 | 142.12.201.1 |
| 132.10.20.0 | 255.255.255.255 | 132.10.4.18 |
| 15.0.0.0 | 255.0.0.0 | 132.1 0.4.1 B |
| 132.10.56.5 | 255.255.255.255 | 132.10.56.4 |
| 132.10.20.5 | 255.255.255.255 | 142.12.201.2 |
| 142.12.0.0 | 255.255.0.0 | Local interfase |
| 0.0.0.0 | 0.0.0.0 | 142.12.201.2 |

Donde (por ejemplo, a cual dirección IP) el enrutador entregará los paquetes direccionados a las siguientes destinaciones? (Use la tabla anterior para responder.)

15.127.243.8
132.10.56.4
132.10.20.3
132.10.20.5
132.10.200.3
37.92.129.1

2.- Considere el siguiente diagrama de una LAN.

Una máquina de prueba (A) es colocada en la LAN y se desea revisar su configuración IP. Se logra esto efectuando el comando ping a la máquina (B) en la misma red. El echo retorna; todo es bien entonces. Entonces intentamos hacerle ping ping a la máquina (C) que se encuentra en otra red. Esta vez el ping falla- ningunos son retornados. Verificamos que mi máquina tiene una ruta desde A a C via el router. ¿Entonces por qué el ping no funciona?

3. En la Compañía X, es hora de decidir en el enrutamiento entre el enrutamiento de la redes de X.

La tarea es dibujar las tablas de enrutamiento para los routers en el diseño de la red de su grupo. Si usted aún no ha asignado algunas clases de IP a la redes en su plan, deberá hacerlo ahora.

¿Percibe usted alguna área de problema /temática?

### Ejercicio 3-3: Examinar las Tablas de Enrutamiento

No se proveen soluciones a este ejercicio. Podemos examinar las tablas de enrutamiento del sistemas usando el comando netstat con la opciones -r o -e:

**$ netstat -r # route -e**

A menudo es útil desplegar los hosts y las redes en la tabla en termino de sus direcciones IP y no sus nombres. Este es muy útil en particular cuando estamos usando servidores de nombre de dominio como es el DNS (lo cual se discutirá más adelante). Para efectuar esta acción, escriba el siguiente comando:

**$ netstat -rn**

Verá una entrada para la interfase de red y también una para la interfase loopback. En esta etapa, podrás ver la pequeña diferencia entre las dos salidas.

Pruebe las opciones -v y -ee del comando route; ¿qué información nos arroja cada una?

Sin una entrada en la tabla de enrutamiento que nos diga como llegar a una red o a un host, no nos podremos comunicar no con la red ni el host. Para ilustrar esto, escriba el siguiente comando:

**$ ping Dirección_IP**

Donde la Dirección_IP es la dirección de uno de los sistemas en otro segmento de la red. Note los errores de mensajes que se despliegan. Ahora ejecute este comando y también observe los mensajes de errores:

**$ telnet Dirección_IP**

## PROTOCOLOS DE ENRUTAMIENTO DINÁMICO

Para comprender las teorías y el funcionamiento de enrutar y los routers debe entender como es que ellos actualizan y mantienen sus tablas de enrutamiento, que es lo que le permite a ellos poder dinámicamente enrutar los paquetes.

En esta sección discutiremos los siguientes tópicos:

- ¿Por qué Enrutar dinámicamente?
- Enrutamiento Dinámico: El Concepto
- Protocolos de Enrutaramiento

- El ICMP y la Redirección de los Mensajes
- La Clasificación de los Protocolos
- Concepto de un Sistema Autónomo
- Protocolos de Interior y Exterior
- Implementaciones UNiX

**¿Por qué Enrutar Dinámicamente?**

Para redes pequeñas, el enrutamiento estático es por lo general suficiente. Muchas redes contienen uno o dos routers . Estas redes pueden ser administradas normalmente usando enrutamiento estático; esto es, las tablas de enrutamiento son entradas a mano y mantenidas manualmente.

A medida que las redes crecen, así incrementa las posibilidad de riesgo. Consideremos una red con cientas de computadoras y docenas de routers; se convierte en casi imposible el mantenimiento y la consistencia de las tablas de enrutamiento.

El propósito de usar enrutamiento Dinámico es para que las redes se actualicen automáticamente. Las redes cambian: nodos se agregan y se eliminan, se crean y eliminan enlaces, a veces temporalmente. Ya que las redes son una colección de computadoras (routers), entonces ¿Por qué es que estas computadoras no puede monitorear y actualizarces para mantener un enrutamiento sincronizado?

**Enrutamiento Dinámico: El Concepto**

En redes a gran escala, la confianza se coloca sobre los sistemas de gateways para que esto mantengan actualizada la información de las rutas que están en todo el Internet. Esta información es potencialmente volátil ya que los enlaces frecuentemente se colocan dentro y fuera de servicio, especialmente en ambiente de redes la área amplia.Y por estos no es práctico, mantener las tablas de enrutamiento de los gateways manualmente.

Cada enrutador propaga la información de routing (o disponibilidad) a sus vecinos. Este también actualiza su tabla de enrutamiento de la información presentada a el por los otros routers. Más a menudo, los gateways implementan lo que es conocido como enrutamiento dinámico para mantener sus tablas correctas. Esto se logra haciendo que los sistemas de gateway ejecuten un proceso especial, conocido como enrutamiento de servidor (routing server) o daemon de enrutamiento (routing daemon).

El trabajo de este proceso es intercambiar información que puede ser usada para determinar las rutas a otras redes. así también como comunicarse con daemons de enrutamiento de otros sistemas, el proceso local también es responsable por el mantenimiento de las tablas de enrutamiento en el sistema local basado en la información que este recibe de los otros sistemas.

De esta manera es mucho más simple mantener las tablas de enrutamiento en un sistema al día con la configuración de la red de Internet.

**El Proceso de Enrutamiento Dinámico**

Es importante entender que enrutamiento Dinámico no cambia la manera en que la capa de Redes efectúa su función de enrutar paquetes, el mecanismo de enrutamiento. Enrutamiento Dinámico determina la información colocada en la tabla de enrutamiento. Si se agrega una nueva red esta es agregada o si un enlace se cae, la tabla de enrutamiento será modificada debidamente. Si se encuentran multiple rutas al mismo destino, la mejor ruta es elegida e insertada en la tabla de enrutamiento. Esta elección se lleva acabo de acuerdo con las políticas de enrutamiento. Cada enrutador contiene un proceso, llamado daemon de enrutamiento,

que efectúa estos cambios (este ejecuta las políticas de enrutamiento).

La información de enrutamiento es propagada de enrutador a enrutador en mensajes de actualización como parte del protocolo de enrutamiento.

**Protocolos de Enrutamiento**

Los protocolos se comunican usando protocolos de enrutamiento. Estos han evolucionado a través de los años. Ahora los podemos clasificar en dos grupos como protocolos de estado de enlace (link-state protocols) y protocolos de vector de distancia (vector-distance). Hay algunos requerimientos principales que deben por lo general ser satisfecho.

**Convergencia**

Esto es para cuando las topología de red cambian. Las tablas de enrutamiento necesitan ser actualizadas para reflejar esos cambios en una forma ordenada. El proceso de actualizar las tablas de enrutamiento cuando las redes cambian debe suceder suficientemente a menudo para que las tablas de enrutamiento reflejen el verdadero estado de la red; con esto se quiere dejar dicho que, las tablas de redes deben converger en el estado de la red rápidamente.

**Consistencia**

Las inconsistencias entre los routers pueden potencialmente convertir una destinación inalcanzable. Un ejemplo es: un router anuncia una ruta que ha sido eliminada desde un router más profundo en la red.

**Seguridad (Autenticación)**

Como los protocolos de enrutar actualizan las tablas de enrutamiento, existen un potencial desafortunado para un atacante desde otra red para enviar una actualización de enrutado a un router. El efecto es que los paquetes no son ya enrutados o pueden ser enrutados al destino equivocado.

**ICMP y la Redirección de Mensajes**

La forma más simple de enrutamiento Dinámico es proveída por el mecanismo de redirección de ICMP. Bajo circunstancias normales, sistemas hosts son inicializados con información minina de enrutamiento. Como descripto anteriormente, un host normalmente sabrá como enrutar paquetes para la red local y también la dirección de un sistema de gateway único al cual todos los paquetes deberán ser enviados.

En algunos casos, este talvez no es la mejor manera de enrutar paquetes. En el ejemplo anterior, al host H1 se le ha dicho que todos los paquetes para otras redes deben ser enrutado a través del gateway G1. Pero, puede ser que la ruta desde Gl a H2 vía G2 es costosa en tiempo o recursos. En este caso, una mejor ruta sería enviar los paquetes por G3, de donde esta puede ser enviada directamente al host H2.

Si la línea entre G1 y G2 está lenta, las tablas de enrutamiento en G1 serán configurada para enviar paquetes para H2 vía G3. Cuando G1 recibe el paquete desde H1 y entonces decide enrutarlo vía G3, este nota que H1 y G3 son las misma red. así que, para evadir este extra salto para paquetes futuro, G1 puede enviar un mensaje especial ICMP de regreso a H1 diciendole que use G3 como un gateway para arribar a H2. Este mensaje es conocido como un mensaje de redirección. Al recibir el mensaje, H1 actualiza su tabla de enrutar para incluir detalles de la nueva ruta.

Use esta técnica, es posible inicializar un host con información minina de enrutamiento, sabiendo que este aprenderá a cerca mejores rutas desde los gateways con el pasar del tiempo.

**Clasificación de Protocolos**

Los protocolos de enrutamiento han evolucionado a través de los años. Observaremos mayormente a dos

tipos:
• Protocolos Vector-Distance
• Protocolos Link-State

Los protocolos difieren en dos maneras. Primero, ellos difieren en el tipo de información que ellos comunican entre los routers. Segundo, ellos difieren en la manera que ellos usan esta información para actualizar las tablas de enrutamiento.

**Protocolos Vector-Distance (Vector-Distancia)**

Los protocolos anteriores vector-distance han sido en muchos casos han sido remplazados por los protocolos link-state. Ejemplos son RIP y Hello. Los routers propagan la información de destinación/cuenta de saltos a los otros routers.

Los protocolos vector-distance esencialmente reenvían (forward/inundación de red) , rutas a través de la red de un router a otro router. Las rutas son enviadas como una tabla, de la cual la primera columna es el destino final (vector) y de cual la segunda columna es la metric (distance). Si el destino no existe en la actualización de la tabla de enrutamiento, una nueva ruta será agregada. Si la lista de enrutamiento actualizada contiene una ruta más corta, la en la tabla es reemplazada. Si la lista actualizada en más larga y el router actualmente enruta vía el router enviando la actualización de enrutamiento, entonces la ruta es reemplazada con la ruta más larga.

RIP es uno de los protocolos de enrutamiento vector-distance más popular. Esto no es debido a cualquier mérito técnico sino debido a que la implementación de RIP fué puesta en disposición a través de la versión de UNiX 4.2 BSD. Esta implementación fué copiada y distribuida ampliamente. De hecho, no hubo originalmente ninguna especificación de RIP. Un RFC fué creado años después, derivado de la implementaciones ya existente.

Para elegir entre rutas alternativas al mismo destino, un valor metric de cuenta de saltos (número de etapas en la ruta) es usado. Las diferencias en tiempo (por ejemplo, entre un enlace rápido y uno lento) puede ser incorporado a través de la escogencia del valor de metric. RIP considera que un valor de más de16 saltos es efectivamente inalcanzable, por esto limitando el tamaño de la red en las que RIP puede ser usado.

Existe una versión nueva de RIP, RIP-2, cual es descripta en el RFC 1388. Este presenta ciertas mejoras sobre el RIP original. En áreas tales como el manejo de las subredes de tamaños diferentes en el mismo host. Este también incorpora algunas nuevas características, así como el soporte para multicasting y una ficha o tag de ruta para hacer que opere mejor con otros protocolos de gateway exterior.

**Protocolos de Link-State**

Un perfecto ejemplo es el OSPF de Dijkstra, algoritmo de Búsqueda, de la red completa de la primera ruta abierta más corta (Open Shortest Path First). Formas más avanzadas de protocolos de enrutamiento usan el método de link-state. Con OSPF, actualizaciones no contienen rutas, solamente el estado de los enlaces directamente conectados.

Hay varias ventajas con enrutamiento link-state, que entre ellas se incluyen:

- Cada enrutador re-calcula sus rutas independientemente y no dependen de equipos externos.
- Enrutadores Link-state garantizan que convergen.
- Mensajes de estado de enlace pasan a través de la red sin ser cambiados, haciendolos así más fácil depurarlos.
- Enrutamiento de Link-state hereda su escabilidad del hecho que el tamaño de los mensajes no depende en el número de enlaces desde el enrutador.
- Enrutamiento de Link-state genera menos tráfico que el enrutamiento de distance-vector porque los men-

sajes son por lo general más pequeños.

Cuando el estado cambia, el router reconstruye su gráfica y vuelve a calcular las rutas usando el algoritmos OSPF de Dijkstra, el calcula la ruta más corta al destino final.

**Otros Protocolos de Enrutamiento**

Uno de los protocolos de interior más nuevo es el ICMP (Router Discovery Protocol). Este sistema provee un mecanismo para que un host elija el enrutador por defecto más apropiado. El protocolo trabaja en un mecanismo basado en anuncios y solicitudes. Cuando un host inicia un software TCP/IP, esta hará un broadcast a un router solicitandole un mensaje. Cualquier enrutador que recibe este mensaje devolverá un paquete anunciante que contiene su dirección IP y un valor preferencial. El host escuchante seleccionará el enrutador con el mejor valor preferencial e instalará ese en su tabla de enrutamiento como el router por defecto.

Además de responder a paquetes de solicitudes, los routers enviarán paquetes de broadcast cada 7-10 minutos. El descanso es al azar entre los envíos de estos paquetes para evitar colisiones con otros routers también anunciandose. Un router puede alterar su valor de preferencia para reflejar el estado de sus intefases. Los hosts escuchantes pueden entonces decidir si otro router es mejor y, entonces, proceder a alterar su entrada de router por defecto. Si un host no recibe un paquete desde su enrutador por defecto cada 30 minutos, eliminará (drop) esa ruta y usaría el próximo mejor router en su lista.

Otro protocolo de interior similar a RIP es el llamado Hello. Este protocolo usa una medida de espera de tiempo (time-delay) para elegir entre rutas alternativas y no una distancia. Este fué el protocolo original en la NSFNET (National Science Foundation Network). Es usado muy poco hoy en día.

Hay varios otros protocolos de enrutamiento que usted puede enfrentar. Ejemplos de estos serian los protocolos OSI, IDRP (Interdomain Routing Protocol) y ISIS (Intermediate System to Intermediate System).

**Conceptos de Sistemas autónomos**

Los sistemas autónomos pueden ser descriptos como una colección de redes y enrutadores debajo de un sólo control administrativo. Internets a gran escala son organizadas en base en una forma jerárquica, con una red central (núcleo) proveyendo un método para la interconexión de las demás redes que la constituyen. La administración de las subsecciones del Internet es a menudo delegada, así reinsertando la estructura jerárquica.

Desde el punto de vista del enrutamiento, es también deseable ver el Internet como una jerarquía de internets más pequeñas.

Cada uno de estos será administrado por una autoridad única independiente, por esto es que son conocidos como sistemas autónomos. Dentro de un sistema autónomo, pueden existir varios otros sistemas más pequeños enlazados por gateways. Estos gateways necesitan saber las rutas entre estas redes en el sistema autónomos pero no necesariamente de los otros sistemas autónomos. Estos son conocidos como gateways de interior.

Cada uno de los sistemas autónomos esta enlazado a la red central “core” por un gateway, cual debe saber por lo menos una información resumida de los otros sistemas autónomos. Estos son conocidos como gateways de exterior.

Para enviar un paquete a un sistema en otro sistema autónomo, un sistema de origen utiliza las tablas de enrutamiento en los gateways de interior de su propio sistema autónomos para llegar hasta en gateway de

exterior. Desde aquí el paquete puede ser reenviado al apropiado gateway de exterior para ser destinado al sistema autónomo donde el gateway de interior puede dirigirlo a su correcta destinación.

Diferentes protocolos para el intercambio de información de enrutamiento entre los sistemas autónomos han sido desarrollado, con uso dependiendo si el gateway en uso es interno o externo. Las características principales de estos protocolos incluyen:

- Libertad de uso de cualquier arquitectura de enrutamiento interno
- Ocultar la estructura interna
- Uno o más enrutadores intercambiando información sobre el alcance de sistemas autónomos a otros sistemas autónomos.

**Gateways Core**

Tradicionalmente los sitios eran conectados a un backbone de Internet. En los inicios del Internet moderno, una red backbone singular central interconectaba todos los sistemas autónomos juntos. Esto significaba que el tráfico entre los sistemas tenían que compartir el único backbone.

La administración del backbone era centralizada y por esto, relativamente fácil de controlar. Pero, este modelo rápidamente confrontó los problemas de escabilidad con todo el tráfico compartiendo un único backbone. Para aliviar el problema de ancho de banda, se agregaron más backbones.

Para mantener el tráfico fluyendo apropiadamente, el intercambio a gran escala de data enrutada tenia que tomar lugar entre los gateways core gateways en el backbone. A medida que nuevas redes fueron agregandose rápidamente, las tablas de enrutamiento crecieron a tamaños inmanejable.

**Peer Backbones**

La solución, la cual aún es la estructura básica del Internet de hoy, fué convertir este modelo de único backbone en una federación de arbitrariamente interconectados sistemas autónomos de tamaños diferentes. Esto simplifica las necesidades de enrutamiento (cada gateway solamente necesita saber acerca de sus vecinos, no todas las rutas posibles) y drásticamente reduce la demanda sobre cualquier de los backbones en particular.

**Protocolos de Interior y Exterior**

Los protocolos de enrutamiento de Internet pueden ser clasificado como interior o exterior, dependiendo si el enrutamiento es dentro o entre las organizaciones.

Protocolos Internos:

- Construir un imagen completa de su sistema autónomo.
- Usar rutas por defecto para tráfico no local.
- Trabajar a un nivel técnico.

Protocolos Externos:

- Trabajar con la información de alcance resumiendo las rutas a todo el sistema en el Internet.
- Puede que tenga que tomar decisiones de enrutamiento por razones no técnicas (otras que no sean de conectividad); Por ejemplo, una corporación con múltiples gateways al Internet probablemente no se desea que sean usados para el tráfico normal.

Protocolos de gateway Exterior permiten que gateways en las fronteras de sistemas autónomos intercambian información de las rutas dentro de los sistemas. Esta información entonces puede ser incorporada den-

tro de las decisiones de enrutamiento tomadas para transferir paquetes de un sistema autónomo a otro.

El protocolo original de ARPANET para esto era el EGP (Exterior Gateway Protocol). Este hay sido suplantado por el protocolo BGP (Border Gateway Protocol), actualmente en la versión 4. BGP puede enrutar paquetes de acuerdo a varias políticas. Estas políticas pueden estar basadas en entras políticas, seguridad o consideraciones económicas. Uno de los objetivos principales de BGP es recortar el monto de tráfico que no es local en este sistema autónomo.

**Implementaciones UNiX**

| | RIP | Hello | OSPF | IS-IS | BGP |
|---|---|---|---|---|---|
| **EGP** | | | | | |
| **routed** | | **X** | | | |
| **gated v2** | **X** | | **X** | | **vl** |
| **gated v3.5** | **X** | **X** | **X** | **X** | **vl,v2,v3,v4** |

Históricamente en los sistemas UNIX, el demonio implementa enrutamienro Dinámico basado en RIP. Cuando comienza, routed inicializa la tabla de información de enrutamiento basado en el archivo /etc/gateways. El formato de este archivo es similar al argumento del comando route.

**[ net | host ] dirección gateway gateway metric [ active | passive ]**

El passive indica que el gateway no difundirá información y, por lo tanto, no serán sacado de la tabla de enrutamiento. Gateways activos, serán removidos de la tabla de enrutamiento si ellos no envían información de enrutamiento.

Los mensaje son transmitidos en cada red, advirtiendo sobre su tabla de enrutamiento cada pocos segundos. El programa también escucha los mensajes que envía desde otros programas routes y actualiza la tabla de enrutamiento local sin información recibida es considerar importante.

El daemon gated es un reemplazo moderno del daemon routed. El daemon de enrutamiento gated soporta mucho más protocolos. En particular, es en un protocolo de enrutamiento Dinámico y esta disponible brevemente.

## Ejercicios 3-4: Protocolos de Enrutamiento Dinámico

En este ejercicio, trabajaremos con preguntas relacionadas con el tema de enrutamiento Dinámico para la Compañía X. No se proveen soluciones para este ejercicio.

1. RIP usa un broadcasts cada 30 segundos para anunciar las rutas disponibles y el número de saltos a ellas. En esta ruta, dos enrutadores están sirviendo tres redes y ambos enrutadores están ejecutando RIP (versión 1). Complete los detalles de las rutas anunciadas y los saltos después que el sistema ha normalizado sus operaciones.

2. Ahora si el enlace de R2 a la Red A se cae. ¿Qué le pasaría a las rutas y los saltos anunciados por ambos R1 y R2 en la reded B y C? ¿Qué más puede pasar?

3. El administrador de Red de Compañía X esta considerando si es apropiado usar enrutamiento Dinámico. El administrador está enfrentando la siguiente situación.
   **La Compañía ha crecido de nuevo y ahora desea conectar otro sitio a su red corporativa. Para hacer esto, el administrador desea colocar otra conexión de 1-Mbps dentro de ambos sitios, así suplementando una conexión existente de 2-Mbps entre ellas. La red WAN debe verse entonces así: IMAGEN AQUI**

La ventajas de este ordenamiento esta supuesto a ser resistente en el caso de fallas de uno de sus enlaces. Enrutamiento puede automáticamente cambiarse.

Usted debe tomar en consideración:

- ¿Vale la pena que la Compañía X considere ejecutar un protocolo de enrutamiento sobre los enlaces WAN?
- Y se es así, ¿cual protocolo(s) pueden ellos ejecutar?
- ¿Tiene la escogencia de un protocolo sobre otro algún beneficio?¿Cuáles son estos beneficios?
- ¿Existe algún beneficio en ejecutar enrutamiento Dinámico dentro de la Compañía X?

**IP MULTICAST**

IP Multicast es una manera efectiva para transportar paquetes desde un origen a multiple destinos. Esto se logra habilitando el origen a enviar paquetes a multiple recibidores Cuáles desean explícitamente recibirlo. Los receptores se hacen parte de un grupo multicast en particular y los paquetes son entregados a todos los miembros del grupo.

El proceso es eficiente por que las aplicaciones IP tradicionales emplean unicasting en la cual una conexión TCP es establecida entre un único emisor y cada uno de los receptores. El emisor replica la data sobre cada conexión, la cual puede usar muchos recursos de red. Aún más, el número de receptores esta limitado por el ancho de banda y el total desempeño se disminuye. El IP Multicast se asegura que las aplicaciones que requieren entrega a multiple destino no causen congestión en la red con copias múltiples de la misma data.

IP Multicast esta basado en el Protocolo UDP (User Datagram Protocol) y por esto, no es confiable ya que el provee un servicio no orientado a conexión o sin estado. Un datagrama multicast es entregado a todos los miembros de un grupo de hosts destinatarios con el mismo esfuerzo y confiabilidad que un datagrama regular del tipo unicast. No hay garantía que los datagramas enviados serán recibidos y no asegura que los datagramas sean recibidos en el orden en que fueron enviados.

Hay tres niveles para que un host participe en IP Multicast. Ellos son:

**Nivel 0** **Un host no puede ni enviar o recibir IP Multicast**
**Nivel 1** **Un host puede enviar pero no recibir IP Multicast.**
**Nivel 2** **Un host puede enviar y recibir IP Multicast.**

Los kernels GNU/Linux de versiónes superior a la 1.1.80 soportan enrutamiento multicast a través de la opción de configuración multicast. Para que un host participe en IP Multicast con niveles 1 y 2, es necesario tener software que permiten al host enviar y recibir datagramas multicast. La membrecía de un host a un grupo es dinámica; los hosts pueden entrar o salir de un grupo en cualquier momento. No hay restricción en el lugar o número de miembros en un grupo de hosts. Un host puede ser miembro de más de un grupo a la vez. Todas estas opciones requieren el router multicast router use el protocolo IGMP (Internet Group Management Protocol) para comunicar la información de membrecía del grupo. El IGMP es un protocolo simple de bajo nivel que permite a los hosts registrarse y recordar el estado de cada grupo multicast que el host pertenece. Para tener multicast habilitado en un sistema GNU/Linux, IGMP debe ser uno de los protocolos IP.

Hay varios protocolos de enrutamiento multicast siendo desarrollado para la toma de decisiones multicast. Entre ellas se incluyen DVMRP (Distance Vector Multicast Routing Protocol), CBT (Core-Based Trees), M-OSPF (La extensión Multicast al Open Shortest Path First), y PIM (Protocol Independent Multicast). PIM esta dividido en dos protocolos: modo denso llamado PIM-DM y modo dispersado llamado PIM-SM. La función principal de este protocolo de enrutamiento es pasar la información de membrecía de grupo entre los

enrutadores multicast y enrutar los datagramas a cada miembro del grupo multicast.

Las función de enrutamiento multicast en un host GNU/Linux es proveído por el daemon llamado mrouted, cual recibe los paquetes unicast encapsulados en una interfase entrante, entonces enruta los paquetes sobre el conjunto de interfases salientes donde quiera que existe el host miembro. Este es similar al daemon routed, pero la diferencia es que routed usa el protocolo de enrutamiento RIP, en donde mrouted usa DVMRP para el enrutamiento multicast. Este daemon es requerido para el enrutamiento de tráfico multicast o proveyendo un tunel a otro sitio que explicamos más adelante.

La configuración de mrouted puede ser cambiado con la modificación del archivo de modificación /etc/mrouted.conf. Existen comandos diferentes para los requerimientos variados.

El comando phyint puede ser usado para deshabilitar enrutamiento multicast en la interfase física. Este comando debe presidir los comandos de tunel, Cuáles son usados para establecer el vínculo de tunel entre una dirección IP local y una dirección IP remota. La opción cache_lifetime determina el monto de tiempo que una ruta multicast capturada en cache permanece en el kernel antes de dar time out. El valor por defecto es de 300 segundos, pero puede ser cambiado a cualquier valor entre 300 (5 minutos) y 86,400 (1 día).

La opción pruning <off/on> es proveída para que mrouted actúe como un servidor de enrutador que no hace pruning. La configuración por defecto habilita a pruning. También es posible iniciar a mrouted en modo de no prunning usando la opción -p en la línea de comandos. Esto solo debe ser llevado acabo para el propósito de prueba solamente. La opción threshold es la mínima IP TTL requerida por un datagrama multicast para ser reenviada a una interfase dada o un tunel. Es usada para controlar el alcance del datagrama multicast y cada enrutador multicast reduce el campo TTL por 1.

La opción rate_limit permite al administrador de red especificar a cierto ancho de banda en Kbps que fueran asiganadas a tráfico multicast.

El comando mrouted responde a las siguientes señales:

**HUP** Esta reinicia mrouted y el archivo de configuración es leído nuevamente cada vez que la señal es invocada.

**INT, TERM** Ambos comandos terminan la ejecución correctamente notificando a todos los routers vecinos.

**USR1** Este da volcado de las tablas internas a /usr/tmp/mrouted.dump

**USR2** Este comando vuelca el cache interno de las tablas a /usr/tmp/mrouted.cache

**QUIT** Volca las tablas de enrutamiento a stderr si el mrouted fué invocado con un valor de debug mayor que cero.

El programa mrouted escribe su pid a /etc/mrouted.pid al inicio por conveniencia de Envío de señales. Este no afecta los protocolos de enrutamiento ya existentes. Un host que ejecuta el daemon mrouted lo hace además de los protocolos estándares de enrutmiento.

El protocolo de enrutamiento estándar ya existente efectúa un tunel multicast en casos donde los datagramas multicast deben pasar a través de routers intermedio que no soportan IP multicast. Para los hosts en redes separadas para comunicarse usando, los routers en ambas puntas deben configurar mrouted usando el comando tunnel. El tunel debe ser establecido en el archivo mrouted.conf de los dos routers que el tunel interconecta. Los routers en ambas puntas del tunel entonces escucha por datagramas que fueron enviados al grupo multicast. Cuando el datagrama multicast arriba en un router pero su destino es el otro router en la otra punta del tunel, mrouted envía el datagrama multicast a mrouted al otro router usando una dirección IP unicast, la cual encapsula el datagrama multicast. En la otra punta del tunel, mrouted extrae el datagrama multicast desde la

dirección IP unicast encapsulada y usa la dirección de destino multicast para reenviar el datagrama a su destinación final.

**RESUMEN**

En este capítulo, fueron introducidos los siguientes tópicos:

- Enrutar (Routing) y los tipos de protocolos de enrutamiento
- Subnetting y supernetting
- Clases de Redes de IP y como sobre pasar sus limitaciones
- Como dividir una red en multiple subnets
- Como incrementar el número de redes y reducir el número de hosts
- Transporte/Delivery de Paquetes
- El rol de las Tablas de Enrutamiento
- Enrutamiento Dinámico
- Filtrado de Paquetes y Firewalls (Corta Fuegos)

- Multicasting

**PREGUNTAS POST-EXAMEN**

Respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿En que se diferencian los conceptos de supernetting del de subnetting?
2. ¿Por qué es que no podemos usar direcciones que terminen en 0 o 255 como direcciones de una subnet?
3. Considerando RIP y OSPF ¿Cúal de estos protocolos de enrutamiento es más avanzado y por qué?
4. ¿Después de cuantos saltos considerará RIP un destino no alcanzable?

# CAPITULO 4

*REDES DEL SISTEMA OPERATIVO GNU/LINUX*

# TCP, UDP, ICMP Y LA CAPA DE TRANSPORTE


| TÓPICOS PRINCIPALES | No. |
|---|---|
| Objetivos | 139 |
| Preguntas Pre-Examen | 139 |
| Introducción | 140 |
| Protocolo de Internet | 145 |
| Formato de los Paquetes IP | 149 |
| ICMP y el Manejo de Error | 152 |
| TCP y la Capa de Transporte | 158 |
| UDP | 161 |
| Resumen | 164 |
| Pregunta Post-Examen | 165 |


## Objetivos

Al finalizar este capítulo, usted estara preparado para efectuar las siguientes tareas:

- Instalar redes TCP/IP y definir los puertos comunes.
- Identificar los protocolos usados para establecer conexiones entre nodos interconectados asi como las convenciones comunes usadas por cada protocolo.

## PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué significa IP?

2. ¿Es el encabezado de IP una fuente principal del sobre peso en la mayoría de implementaciones de IP?

3- ¿Cuáles tres campos son contenidos en el cabezal IP?

4. ¿Cuáles números de puertos son reservados para servicios bien conocidos?

# INTRODUCCIÓN

En este capítulo cubriremos la capa de Transporte de la pila del protocolo TCP/IP. Esto incluye el TCP (Transmission Control Protocol), el cual maneja las comunicaciones orientadas a conexión y el UDP (User Datagram Protocol), el cual maneja la comunicación sin conexión. El ICMP (Internet Control Message Protocol) es usado como una capa de control para permitir tráfico IP entre los sistemas para fluir libremente. También cubriremos formatos de tramas y fragmentación.

# PROTOCOLO DE INTERNET

El Protocolo de Internet (IP) es un protocolo sin conexión, orientado a datagramas y reenviados de paquetes. La Data se pasa a IP desde los protocolos más alto de transporte para entrega. El IP envía un paquete en la próxima etapa de su viaje hacia su destino final basada en la dirección IP de destino y la información de enrutamiento que se encuentra almacenada internamente dentro del protocolo. Un paquete IP puede que pase por varios gateways o saltos, antes de alcanzar su destino final.

A menudo el IP es describido como un protocolo no confiable. Esto no significa que no sea útil. Sino, esto significa que su entrega no es garantizada, IP depende de entrega del mejor esfuerzo; los paquetes en esta capa no son confirmados.

El hecho que IP no garantiza la entrega no es de ninguna gran preocupación. Ante todo, IP asume que las capas más altas (como es la de la capa de Transporte) serán las responsables de proveer la confiabilidad. Segundo, IP fué diseñado para ser simple, eficiente y fácil de implementar. Se da muchas veces el caso que todo lo que es requerido es que sea rápido fast y la entrega de mejor esfuerzo.

Piense en no confiable al igual que enviar un correo tradicional. Una vez enviado, usted no puede estar seguro si será entregado a su destino final o si llegará en su totalidad. Pero sabemos, que basada en la confiabilidad de del servicio postal haya un buen chance de que nuestro correo llegue a su destino y que el tiempo que tome dependerá de la distancia que nuestro correo tenga que viajar (y la eficiencia de cada una de las diferentes agencias postales que nuestra carta tendrá que viajar para llegar a su destino). Asi que, el servicio postal e un mecanismo de entrega basado en mejor esfuerzo, y mecanismos de entrega no confiable, pero este no es un problema mayor que no nos permita usarlo y esta basado en el nivel de confiabilidad que este provee.

Si estas inseguro de como se puede proveer un servicio confiable en un nivel más alto que usa IP (un servicio no confiable) en un nivel más bajo, consideremos como podemos operar un servicio de transferencia de data que garantice confiabilidad usando el servicio postal. Una idea que surge es la de establecer un mecanismo en el cual se establezca un método de confirmación de devolver una postal al emisor original. Si usted envía una tarjeta y después recibe una confirmación para esa tarjeta, usted puede estar seguro que la original llegó con éxito.

### Los Protocolos

El sistema TCP/IP esta basado en un conjunto de protocolos en capas. Esto permite implementaciones en cada capa ser separadas de cada una esto hace que el sistema completo sea más modular y más fácil de depurar.

El TCP/IP contiene dos protocolos básicos para la transmisión de data estos 2 protocolos son UDP y TCP, en el modelo OSI, ellas residen en la capa de transporte. Además, existen el ICMP, el cual se encuentra en la capa de red.

**ICMP**

Localizado en la capa de red del modelo OSI es el protocolo de control de mensajes del Internet (Internet Control Message Protocol). El propósito del ICMP es proveer un mecanismo de control de error al IP. Los mensajes ICMP son de dos tipos básicos: Control de error y consulta. Algunos mensajes ICMP de control de error son:

| | |
|---|---|
| 3 | Destino inalcanzables |
| 4 | Source Quench |
| 5 | Redirección |
| 11 | Tiempo Excedido |
| 12 | Problema de Parámetro |

Los mensajes ICMP usado para consulta son:

| | |
|---|---|
| 0 | Echo Reply/Repuesta |
| 8 | Echo Request/Petición |
| 9 | Router Advertisement/Anuncio |
| 10 | Router Solicitation/Solicitud |
| 13 | Timestamp Request/Petición |
| 14 | Timestamp Reply/Repuesta |
| 17 | Address Mask Request/Petición de Mascara de Dirección |
| 18 | Address Mask Reply/Repuesta de Mascara de Dirección |

**UDP**

El Protocolo de Datagrama de Usuario (User Datagram Protocol) es un protocolo de la capa de Transporte utilizado principalmente para comunicaciones cortas y de una sola vía. Los datagramas UDP no contienen información acerca de los otros paquetes. Asi que, cada paquete UDP es independiente uno del otro, cada mensaje no puede ser mayor en tamaño al máximo permitido por el paquete UDP.

Otro aspecto del UDP es su falta de control de error. La única forma de implementar algún tipo de control de errores es a traves del uso de checksum. Si el checksum no computa apropiadamente, el paquete es dejado caer, el UDP depende del ICMP para enviar los mensajes apropiados de error.

Puede ser que el UDP no le parezca muy útil, pero, este provee una magnifica alternativa para las comunicaciones que sólo necesitan enviar pequeños mensajes sin el alto costo fijo de las comunicaciones a gran escala usada por el TCP.

**TCP**

Debido a la falta de funcionalidad ofrecida por el protocolo UDP, para las comunicaciones que requieren más confiabilidad, el TCP (Transmission Control Protocol) es usado. TCP es un protocolo más robusto que el protocolo UDP; este provee control de error, flujo de data y comunicación de dos vía.

Las conexiones TCP administran las comunicaciones de dos vía y manejan errores a traves de opciones (flags) en el cabezal del paquete TCP. Las dos opciones más comunes del TCP son sincronizadas (SYN) confirmación (ACK). SYN es enviado al principio de una conexión y ACK es enviado una vez el paquete ha sido recibido por el host destinatario.

Una conexión empieza con el equipo cliente enviando una información de inicializacion la máquina servidor en un paquete con la opción SYN establecida. El servidor entonces reconoce la información de inicialización del cliente con una opción ACK y envía su propia información de inicialización con la opción de SYN establecida. El cliente entonces reconoce los paquetes SYN ACK del servidor con otro paquete ACK. Este proceso de inicialización es lo que es conocido como el saludo de tres vía (three-way handshake).

El control de error es implementado en el TCP a traves de la opción ACK. cada paquete enviado por TCP es enumerado y un temporizador es iniciado cuando cada paquete es enviado. Si una confirmación no es recibida antes de que se venza el temporizador, el paquete entonces es reenviado.

El flujo de data es acomodado a traves del cabezal del paquete. Como TCP enumera cada paquete, el número del paquete es colocado en el cabezal del paquete TCP. El emisor entonces, puede enviar multiple paquetes consecutivamente. Cuando cada paquete arriba, este es colocado en la cola y entonces los paquetes recibidos pueden ser colocados en el orden apropiado.

## FORMATO DEL PAQUETE IP

El campo Header Length (Longitud del Cabezal) específica la longitud del cabezal (header). La longitud del header se mide en words de 32-bit, dando un tamaño máximo del cabezal de 60 bytes. Por lo general el Header Length es 5.

El campo Versión lleva la versión actual del IP, la cual hoy es la 4. La próxima versión de IP, la versión 6, será la próxima generación IPng, esta actualmente siendo actualmente especificada.

El campo Service Type (Tipo de Servicio) es usado por algunos enrutadores al tener que decidir como reenviar el datagrama. Este puede indicar que el datagrama debe ser enviado via el router más barato, rápido y más confiable o por de más ancho de banda. Un datagrama puede ser enviado sin la opción de Service Type establecida.

La Total Length (Longitud Total) es el tamaño total del datagramas en bytes, incluyendo el cabezal. Esto nos da el tamaño máximo del datagrama de 65,536 bytes, o 64 KB.

Los campos Identification, Flags y Fragment Offset son usados para controlar, si es necesario, la fragmentación y el reemsamblaje de los datagramas. De esto hablamos más adelante en este mismo capítulo.

El campo Time to Live (Tiempo Restante) es un contador que es reducido cada vez que un paquete cruza el gateway. Cuando el contador llega a '0', el paquete es descartado. Esto es para prevenir que datagramas que están equivocadamente enrutados congestionen la red si viven un tiempo indefinido.

El campo Protocolo (Prorocol) indica cual protocolo de transporte es que va a recibir el paquete en la dirección de destino. Todos los protocolos que usan IP (TCP, UDP, ICMP, etc.) llevan un checksums de su propio cabezal y data. Asi pueden evitar redundancia, el Header Checksum cubre solamente el cabezal IP. Si la capa de IP receptora detecta un datagrama corrompido, este lo descartará.

El datagrama será entregado a la dirección especificada en el campo 'Destination Address' del encabezado. Si este no se encuentra en la misma red como el sistema actual, entonces el datagrama es reenviado a un gateway apropiado que le dará el destino final. Esta información de enrutamiento es almacenada en una tabla por separado. Además, la dirección de origen 'Source Address' del datagrama esta incluida en su propio campo.

Las Opciones IP pueden ser usadas para gobernar el enrutamiento (enrutamiento estricto o liviano) de los datagramas, establecer cualquier restricciones de seguridad y manejo o grabar las direcciones y la fecha de los enrutadores por los cuáles los datagramas pasaron para arribar a su destino final. Algunos enrutadores no examinan estas opciones.

En esta sección discutiremos los siguientes tópicos:

- IP Checksum

- Tipo de Servicio
- Fragmentación y Reemsamblaje
- Tiempo de Vida/Time to Live
- Opciones IP

Calcular el cheksum del encabezado IP es causa de costos considerables en las implementaciones IP. Por estas causa es que la nueva versión de IP, la versión 6, ha removido el checksum. ¿Cómo, entonces, podemos garantizar la integridad del encabezado IP? (Cubriremos este tópico más adelante.)

## Tipo de Servicio

Los bits del campo de tipo de servicio (TOS) son un conjunto de cuatro indicadores de un bit de la cabecera de IP. Si uno de estos indicadores de bit vale 1, un encaminador puede manipular el datagrama de forma diferente del caso en el que ningún indicador valiera 1. Cada uno de los cuatro bits tiene un propósito diferente y sólo uno de los bits de TOS puede valer 1 al mismo tiempo, es decir, las combinaciones no están permitidas. Estos indicadores de bit se denominan de "tipo de servicio" porque permiten que la aplicación que transmite los datos informe a la red del tipo de servicio de red que requiere. Los bits del ToS le permiten a la máquina que envía requerir que los enrutadores manejen los paquetes de cierta manera. Una manera posible es precedencia. Esto permite para la prioridad de paquetes, aunque por lo general no es implementado en los enrutadores. Típicamente, ToS permite que un paquete sea enrutado basado en uno de lo siguiente:

**Demora mínima/Minimal Delay**

Se utiliza cuando se le da la máxima importancia al tiempo de viaje de un datagrama del 'host' de origen al 'host' de destino (demora). Por ejemplo, un suministrador de red podría estar utilizando tanto conexiones de red de fibra como por satélite. Los datos transportados por las conexiones por satélite tienen que viajar más lejos y su demora entre los mismos extremos será por lo general mayor que la de las conexiones de red terrestres. Un suministrador de red podría elegir asegurarse que los datagramas con este tipo de servicio no se transporten por satélite.

**Rendimiento máximo/Maximal Throughput**

Se utiliza cuando el volumen de datos transmitidos en cualquier período de tiempo es importante. Existen numerosos tipos de aplicaciones de red para las que el tiempo de demora no es muy importante pero el rendimiento sí que lo es; por ejemplo, las transferencias de ficheros en bloque. Un suministrador de red podría elegir encaminar los datagramas con este tipo de servicio vía rutas de demora alta, pero de gran ancho de banda, como las conexiones por satélite.

**Fiabilidad máxima/Maximal Reliability**

Se utiliza cuando es importante tener alguna certeza de que los datos llegarán al destino sin necesidad de una retransmisión. El protocolo IP puede transportarse sobre un número variado de medios de transmisión de bajo nivel. Mientras que SLIP y PPP son adecuados para protocolos de enlace de datos, no son tan fiables para transportar IP como otras redes, como las redes X.25. Un suministrador de red podría tener disponible una red alternativa, que ofreciera alta fiabilidad para transportar IP y que se utilizaría cuando se eligiera este tipo de servicio.

**Coste mínimo/Minimal Cost**

Se utiliza cuando resulta importante minimizar el coste de los datos transmitidos. El alquiler de ancho de banda de un satélite para una transmisión transoceánica cuesta generalmente menos que el alquiler de espacio de un cable de fibra óptica sobre la misma distancia, por lo que los suministradores de red pueden elegir proporcionar ambos y cobrar de forma diferente según sea el que se utilice. En este escenario, el bit de tipo de servicio de "coste mínimo" puede ocasionar que los datagramas sean encaminados vía la ruta de menor coste por satélite.

## Fragmentación y Reensamblaje

Cuando la capa de Transporte pasa los data al IP, esta será organizada dentro de un datagrama IP. Si el datagrama creado es más grande que el MTU, Unidad Máxima de Transmisión (Maximum Transmission Unit), de la red, el IP dividirá la data en partes más pequeñas llamadas fragmentos. Estos fragmentos son encapsulados en en paquetes más pequeños, los cuáles son enviados por la red al host receptor y

reensamblado.

Como cada fragmento es un paquete IP individual, cada uno de estos paquetes puede ser enrutado a su destino cía tres rutas diferente. Esto significa que al único host que todos los fragmentos son garantizados a arribar es el host receptor. Asi que, el reensamblaje sólo toma lugar en el host de destino.

La fragmentación original toma lugar basada en el MTU del host que envía. A medida que los fragmentos son enrutados a su destinación, el MTU puede puede variar de una red a la otra. Esto quiere decir que si un datagrama que contiene un fragmento confronta una red con un MTU inferior, el fragmento mismo puede ser que sea aún más fragmentado. Esto se maneja bien por IP debido a que cada fragmento contiene información de como (offset) recomponerse del principio del datagrama original.

Si la capa de Transporte desea asegurarse de que un datagrama no este fragmentado, le puede instruir a IP establecer la opción (flag) de NO fragmentar en el cabezal del datagrama. Si el cualquier punto del viaje cualquier datagrama es más grande que MTU, este será descartado. En este caso, ICMP causará que un mensaje de error sea generado.

Aquí le presentamos un ejemplo de fragmentación:

El cabezal IP contiene tres campos para asistir con el proceso de fragmentación: Identification/

| **Host A** | 4,000 byte UDP datagrama<br>Ethernet: MTU - 1,500 bytes de data | | | **Host B** |
|---|---|---|---|---|
| | **Datagrama Original de IP** | **Fragmento 1** | **Fragmento 2** | **Fragmento 3** |
| **Identification** | 47987 | 47987 | 47987 | 47987 |
| **Fragment Offset** | 0 | 0 | 1480 | 2960 |
| **Size** | 4,000 | 1480 | 1480 | 1040 |
| **Flags** | None | More Frags | More Frags | none |

Identificación, Fragment Offset/Posición del Empalme y Flags/Opciones.

Cada datagrama IP tiene un campo único de Identificación dentro de su cabezal. Este es un campo de un número de sólo 16-bit mantenido por el host emisor, el cual es incrementado por cada datagrama que este envía (sin incluir los fragmentos). Si un datagrama está fragmentado, cada fragmento compartirá el número de Identificación original del datagrama. Esta es la forma por la cual el host receptor sabe a cual datagrama en particular pertenece un fragmento.

El Fragment Offset ofrece el punto de empalme desde el datagrama de origen al fragmento actual. Asi es que el receptor sabe donde cada fragmento cae en el datagrama original al momento de reemsamblaje.

El campo Size/Tamaño de cada fragmento contiene el tamaño del paquete. La suma del tamaño del total de todos los fragmentos será el tamaño del datagrama original. El campo Size muchas veces también es referido como el campo Longitud Total (Total Length).

haya una opción/flag en particular establecida en el campo Flags para todos los fragmentos excepto el último. La opción more fragments es de la manera que el receptor reconoce de esperar por más fragmentos

antes de empezar a reensamblar el datagrama. A primera vista aparenta que el último paquete de fragmento pudiese ser confundido con un datagrama normal, un datagrama sin fragmentar. Pero, el software IP receptora encontrará que el Fragment Offset no es igual a cero y por eso identificará el último fragmento del datagrama. Esto es importante porque los fragmentos pueden ser enrutados por separado, los fragmentos no siempre arribaran en el mismo orden en el cual fueron enviados.

Si todos los fragmentos de un datagrama no son recibidos dentro de un período de tiempo, normalmente 30 o 60 segundos, el fragmento recibido será descartado, igual que si el datagrama nunca hubiese llegado. No haya manera de reenviar solamente un fragmento de un datagrama. Si el fragmento fué producido por un enrutador intermediario, el host que envía ni sabe si el datagrama que el el envía es fragmentado o no. Esto significa que el datagrama por completo deberá ser retransmitido por completo. Por esta razón, la fragmentación es muy amenudo evitado.

Los protocolos de la capa de Transporte manejan la fragmentación en diferente maneras. El TCP trata de evitar la fragmentación reduciendo el tamaño de sus segmento. En algunas aplicaciones UDP, es posible limitar el tamaño de los datagramas pasados a IP.

**Tiempo de Vida/Time to Live**

Específica el tiempo(en segundos) que se le permite viajar a este datagrama. Cada "router" por el que pase este datagrama ha de sustraer de este campo el tiempo tardado en procesarlo. En la realidad un "router" es capaz de procesar un datagrama en menos de 1 segundo; por ello restará uno de este campo y el TTL se convierte más en una cuenta de saltos que en una métrica del tiempo. Cuando el valor alcanza cero, se asume que este datagrama ha estado viajando en un bucle infinito y se desecha. El valor inicial lo debería fijar el protocolo de alto nivel que crea el datagrama.

Bajo ciertas circunstancias, como es por ejemplo un enrutador mal configurado, los paquetes pueden potencialmente viajar en un bucle para siempre. El efecto de este mecanismo puede ser visto desde el utilitario traceroute.

**Opciones IP**

Las Opciones de IP permiten al host emisor requerir cierta condiciones de enrutamiento o otros servicios de los routers. Estudiaremos a cuatros principales: Record Route, Timestamp, loose-source routing, y strict-source routing. Al usar la opción Record Route esta nos permite insertar la ruta que el paquete viajó dentro del cabezal IP, escribiendo en secuencia la direcciones IP de todos los routers que el paquete atravesó.

Desafortunadamente esto no es tan útil como se ve, ya que haya in espacio limitado en el cabezal para almacenar estas rutas. Puesto en práctica, para almacenar la ruta de los paquetes, debe usar un mecanismo basado en TTL similar al de traceroute. haya equipos de enrutar que no soportan Record Route.

En su viaja a traves del Internet, un paquete puede cruzar 20 y hasta 30 routers desde su dirección de origen hasta su destino final. Esto excede por mucho la capacidad del mecanismo de Record Route.

**Timestamp/Sello de Tiempo**

El Timestamp fué concebido como una extensión del mecanismo de Record Route. La hora y la fecha que un paquete pasa a traves de un router es almacenado, además de su dirección IP. Desafortunadamente, esto reduce el espacio disponible aún más, asi que el Timestamp es aún más limitado que Record Route.

peor aún, el tiempo que es sellado es el del router local. El tiempo entre dos routers no necesariamente

tiene que estar sincronizado, al menos que un protocolo como el Network Time Protocol (NTP) se este ejecutando. Asi que los tiempos reportados de retraso pueden ser no actualmente ciertos sino afectado por falta de sincronización. Vuelvo y repito, esta opción no siempre esta disponible en los enrutadores más modernos por falta de popularidad de su uso.

**Source Routing/Origen del Enrutamiento**

También es posible especificar por adelantado que ruta debe seguir un paquete. Esto se puede lograr de dos maneras:

- **Loose-source routing/Enrutamiento Estricto**
  Aquí le especificamos en el cabezal del IP una lista de enrutadores por los cuáles el paquete deberá atravesar. El paquete también puede viajar vía otros routers que no estén listados en la lista pero sólo será exitoso en alcanzar su destino final si lo hace vía esos routers listados.
- **Strict-source routing/Enrutamiento Flexible**
  En este escenario la lista en el cabezal explícitamente específica cuáles enrutadores el paquete deberá atravesar. No es permitido viajar vía cualquier otro enrutador que no este listado.

Enrutamiento Flexible y Enrutamiento Estricto (Loose- and strict-source routing) fueron conce para proporcionar una medida de control sobre la ruta que los paquetes toman. Pero, como las otras opciones, estas on muy dificiles de poner en practica. Muchos enrutadores modernos se reusan aceptar paquetes con las opciones de source-routed aplicadas.

En la practica, el control del enrutamiento está basado en el uso de políticas de enrutamiento dinámico (presentaremos más información sobre esto más adelante) y la cooperación mutua entre los administradores de sistemas de las diferentes organizaciones que el tráfico de sus paquetes transversan.

## MANEJO DE ERROR ICMP

El IP no tiene mecanismo interno del manejo o reporte de errors. Aunque esto tiene como ventaja que el protocolo es más simple, esto significa que los errores enfrentados con el transporte y enrutamiento no serán anunciados y permanecerán sin ser corregidos. El ICMP y el IP son virtualmente lo mosmo excepto por unas cuantas pequeñas diferencias:

- ICMP agrega el reporte de error y otros mensajes de control a IP.
- ICMP también reside en la capa de Red y trabaja en conjunto con el IP. Usa el IP como agente de transporte.
- Los mensajes ICMP son usados para indicar errores como cuando los hosts no son alcanzables, tiempo de espera agotado o timeouts, o error en el formato del datagrama. También pueden ser usados para el control del flujo cuando una destinación nos informa que no puede aceptar más datagramas (source quench). El ICMP puede ser usado para requerir información como son los timestamps (para la sincronización del reloj) o simplemente una respuesta a un petición de echo (echo es usado para probar conexiones, como en el comando ping).
- Con la excepción de las peticiones y respuestas de echo del ICMP, los mensajes ICMP no son enviados en respuestas a otros mensajes ICMP. Ellos tampoco son enviados en respuestas a los datagramas de respuestas. Esta es una característica importante que evita la posibilidad de tormentas de broadcast, en la cual mensajes son intercambiados entre los hosts, lo cual consumirá mucho ancho de banda.

Los siguientes tópicos son discutidos en esta sección:

- Formato del Cabezal ICMP
- Tipos de Mensajes ICMP

**Formato del Cabezal ICMP**

Cada mensaje ICMP es encapsulado en un datagrama IP. El mensaje tiene cuatro partes:

- Los tipos de mensajes. Existen quince tipos diferente de mensajes ICMP, descritos más´s adelante en esta

misma sección.

- El código del mensaje es en efectivo un subtipo. Por ejemplo, haya un tipo para el error “destination unreachable”. Existen dieciséis valores de códigos diferente para este tipo, las cuáles son establecidas por el emisor para decirle al receptor porque el paquete no pudo ser entregado.
- El checksum cubre ambos el cabezal y la data.
- La data difiere dependiendo en el tipo y el código del mensaje en el datagrama. Por ejemplo, todos los mensajes de error contienen el cabezal IP del datagrama erráneo y las primeras 8 bytes de su data, la cual contendrá parte del cabezal del protocolo del datagrama enviado. Este contiene el número del protocolo para que el receptor del mensaje ICMPsepa a quien informar del error.

**Tipos de Mensajes ICMP**

Los quince tipos de mensajes ICMP son como sigue:

| Tipo | Uso |
|---|---|
| 8 | Petición de Echo (enviado por ping) |
| 0 | Repuesta de Echo (recibido por ping) |
| 3 | Destino inalcanzable (no puede entregar o reenviar un datagrama) |
| 4 | Control de flujo (Source quench) |
| 5 | Redirección (alterar tablas de enrutamiento en el receptor) |
| 9 | Anuncio de Router |
| 10 | Solicitud del Router |
| 11 | Tiempo Excedido (TLL igual a 0) |
| 12 | Parameter problem (invalid option in IP header) |
| 13 | Petición del Sello de Tiempo/Timestamp (usado para sincronizar de la red) |
| 14 | Respuesta al Sello de Tiempo/Timestamp (usado para sincronizar de la red) |
| 15 | Petición de Información (obsoleto) |
| 16 | Respuesta de Información (obsoleto) |
| 17 | Petición de mascara de dirección |
| 18 | respuesta de mascara de dirección |

**TCP Y LA CAPA DE TRANSPORTE**

En esta sección cubriremos las capas encima del IP (Internet Protocol). Discutiremos el primero de dos protocolos de transporte del TCP/IP- Transmission Control Protocol (TCP). En esta sección también se incluyen algunas aplicaciones que usan este protocolo, el concepto de sockets y número de puerto y el mapeo de estos números de puertos a servicios.

Los siguientes tópicos son discutidos en esta sección:

- ¿Por qué tener aún más Capas?
- Interfase a la Capa de Transporte
- Conceptos
- TCP Fundamentos
- TCP Formato de la Trama
- TCP Opciones
- TCP Control de Flujo
- SYN Cookies
- Puertos y Sockets
- Pasar Data a Aplicaciones
- HTTP
- Servidores GNU/Linux

**¿Por qué tener aún más Capas?**

El IP es excelente en lo que hace, pero este fué diseñado para un propósito específico y no ofrece todo lo que necesitamos para un ambiente de aplicaciones distribuidas en red. Aquí le presentamos una lista de las piezas que aún faltan para completar el rompe cabeza:

- Una manera de conversar con una aplicación específica en un host
- Un mecanismo confiable de transferencia de data
- Una manera de enviar flujo continuo de data
- Control de Flujo de data

Pero, afortunadamente estas piezas que faltan son exactamente lo que la capa de Transporte fué diseñada a enfrentar. Ahora podemos observar alguno de los temas y maneras que esto puede ser resuelto.

**Interfase a la Capa de Transporte**

El IP es usado por los dos mayores protocolos de Internet de la capa de Transporte, el TCP y UDP, para transmitir data sobre la red. La porcion de data de un datagrama no contiene indicación de cual protocolo se encuentra operando. Pero, el cabezal IP si contiene un campo Protocol Number (Número de Protocolo) que es usado para especificar cual protocolo de transporte he usado para enviar la data. En el destino final, el IP puede pasar data al software apropiado del protocolo de transporte.

El número del protocolo es también usado para identificar los paquetes ICMP y lo reenvía al software del protocolo ICMP para procesarlo, aunque este no es estrictamente un protocolo de transporte.

**Conceptos**

Todo lo que la capa de Red provee es la transmisión de paquetes de IP de host a host a traves de la red. Esta no tiene conocimiento alguno de cuáles aplicaci son propietarias de cual data pero simplemente se le canalizan toda la data para un host en particular de punta a punta en la red.

En la capa de Transporte es donde se comienza a fragmentar la data y a direccionarla a y desde las aplicaciones a traves de la red. Como veremos en esta sección la próxima, existen dos tipos de flujos disponibles: los confia y los no confiables.

**Protocolos sin Conexión**

Imaginemosnos las tramas de data como una carta postal. Durante su viaje por diez ciudades, usted envía una carta postal de cada ciudad que visita a un amigo en otra ciudad. El envío es hecho a traves de un protocolo no confia (como es el servicio de correo) lo que significa que no haya garantía alguna que las diez cartas llegarán a las manos de su amigo en la otra ciudad. Aunque el servicio no es garantizado por lo general las cartas enviadas por el servicio postal llegarán a su destino final, tampoco no haya garantía de que llegaran en el orden que se enviaron.

El IP trabaja de esta misma manera. Los paquetes IP pueden perderse o hasta dañarse. No haya un mecanismo en la capa de Red para manejar las notificaciones de la perdida al emisor o para requerir que el paquete sea reenviado. Estas funciones se dejan a un protocolo de más alto nivel.

Existen maneras de combatir este problema si se requiere oriniento correcto y confiabilidad. Por ejemplo, podemos asegurarnos que nuestro amigo en la otra ciudad envíe una carta de reconocimiento de que recibió nuestra carta. Entonces, un tiempo acordado podemos reenviar cualquier carta que no recibimos una carta de confirmación de el. Una vez se han recibido todas las cartas de confirmación, ya sabemos que todas nuestras cartas fueron recibidas.

También, podemos enumerar las cartas cuando las envió y entonces nuestro amigo puede usar estos números para ponerlas en orden y reconstruir todo el viaje por las ciudades de las diez postales.

Eso si, ambos de estos métodos tienen presentan problemas. Ahora cada vez que envío data debo esperar por una confirmación . La pregunta entonces es, ¿Qué tiempo debo esperar? Todo este mecanismo restará velocidad al proceso de la transmisión de la data. Si la transferencia es poca, funciona, pero para transferencias de data a alta escala, necesitamos un mecanismo diferente.

**Protocolos Orientados a Conexión**

Los protocolos Connection-oriented establecen una conexión con el otro punto, transfieren la data, luego cierran la conexión. La conexión en dos vía (full duplex) es analoga a la conversación por el teléfono. Los protocolos orientados a conexión son conocidos como confiable porque ellos incluyen una garantía de entrega de la data, asi previniendo la perdida de información y la data llega al otro punto en el mismo orden en que fué enviada.

Considere el intercambio de data como una conversación telefónica. Primero, la conexión es establecida, e.j., se marca el número y se conecta. Entonces el intercambio de la data en ambas direcciones comienza. Todo el tiempo, la data arriba al otro punto en el mismo orden en que fué enviada; esto es que sus palabras no llegan en un orden diferente al que usted la habló. Cuando la transferencia (en el ejemplo la conversación) es completada, la conexión es cerrada; y ambos cuelgan.

Este mecanismo es completamente confiable ya que no haya manera de perdida de data sin nuestro conocimiento. Este mecanismo es el que necesitamos en la transferencias de data a gran escala.

**Fundamentos de TCP**

El primer protocolo de transporte en el suite TCP/IP es el TCP. El TCP es muy sofisticado, garantizando que la data es entregada de punta a punta en una manera completamente confiable, en orden y sin duplicación. El TCP es un protocolo orientado a conexión. Lo que significa que una conexión logical, a veces conocida como un circuito virtual, debe ser establecido entre los dos puntos antes de que la data pueda ser transmitida o recibida. Esto es muy parecido a una llamada telefónica en la que uno debe marcar el número de teléfono de otra persona y alguien debe responder el otro teléfono antes de que podamos empezar a conversar.

Una vez establecida la conexión, el TCP garantiza que la data será entregada a su destino sin perdida ni duplicación. Cada porción de data, llamada un segmento, es transmitida con un checksum que puede ser utilizada para asegurarse que la data arrió intacta. El emisor espera que cada porción de data será confirmada su entrega y retransmitirá si no recibe una confirmación en un período acordado. Cualquier paquete que ha tomado más de 1,500 más para arribar a su destino final es considerado sospechoso y es descartado. Además, un paquete que ha efectuado más de 16 saltos o hops es considerado no confiable. Desafortunadamente, el Internet ha crecido tanto que es muy difícil encontrar un paquete que ha sobrevivido más de 10 saltos y que un paquete tome los 16 saltos permitidos es muy común.

El tamaño de un segmento es determinado por el software del TCP. Normalmente, una opción optima es hecha de acuerdo con la red subyacente. Por ejemplo, la elección para Ethernet es de 1,024 bytes ya que esta permite que cada segmento quepa tramas Ethernet individuales.

El TCP también permite el control de flujo fuera del mecanismo primitivo de source quench del ICMP a traves del uso del protocolo de sliding window. (lo veremos más adelante en este mismo capítulo en Control del Flujo TCP).

El TCP proveé una vista basada en flujo de bytes de las aplicaciones de comunicación. En el punto de recepción, el software TCP reensambla los segmentos en el flujo de data y lo pasa al software al nivel de aplicación. No existe fronteras lógicas en la data que se envía o l aquea se recibe, a diferencias del UDP, el cual

recibe y envía unidades discretas.

### Formato de la Trama TCP

El TCP soporta full-duplex, comunicación orientada a flujo de bytes entre los procesos. Pero para la transmisión, los flujos de data deben ser divididos en segmentos que puedan ser encapsulados en datagramas IP. A cada segmento se le adhiere información de control y es entonces transmitida en un datagrama IP.

El cabezal TCP contiene varios campos importantes, incluyendo:

- **Source Port (16 bits)**
  Este campo identifica el puerto de origen. Cuando un cliente hace un petición a un servidor, el puerto de origen puede ser visto como el número de puerto de la aplicación cliente para la conexión inicial.
- **Destination Port (16 bits)**
  Este campo identifica el puerto de destino. Cuando un cliente hace un petición al servidor, el puerto de destino puede ser visto como el número de puerto de la aplicación servidor para la conexión inicial.
- **Sequence Number (32 bits)**
  Este campo indica el número de secuencia del primer byte de data en el segmento excepto cuando esta presente la opción SYN (este flag que hace referencia a un número sincronizado de secuencia). Si SYN esta presente, el número de secuencia es seleccionado al azar. El emisor y el receptor sincronizan el número de secuencia durante el establecimiento de la conexión.
- **Acknowledgment Number (32 bits)**
  Esta es una confirmación adicionada que contiene el número de la secuencia del próximo octeto que la entidad TCP está a la espera de recibir.
- **Header Length (4 bits)**
  Este campo específica el número de palabras de 32-bit en la cabecera. El tamaño máximo del cabezal TCP es de 60 bytes, pero el tamaño normal, sin las opciones es de 20 bytes. El cabezal TCP no contiene un campo Length. La longitud del segmento es calculado desde la longitud el datagrama IP y el HLEN TCP.
- **Reserved (6 bits)**
  This field is reserved for future use.
- **Flags (6 bits):**
  - **URG** el puntero urgente. Indica que data ha sido colocada en e flujo de data.
  - **ACK** es una confirmación. Identifica información de la confirmación en el packet.
  - **PSH** es la función push. Forza a TCP a liberar la data.
  - **RST** reinicia la conexión. Rápidamente desactiva una conexión TCP.
  - **SYN** sincroniza un número de secuencia. Este bit se establece en 1 para los primeros dos paquetes para las conexiones que usan TCP en la capa de Transporte.
  - **FIN** significa que el emisor no transmitirá más data. Es la manera normal de finalizar una conexión TCP.
- **Window (16 bits)**
  Este es el asignador de crédito al control de flujo, en bytes. Contiene el número de octeto de data, empezando por el indicado en el campo Acknowledgment, que el emisor está dispuesto a aceptar.
- **Checksum (16 bits)**
  Este es usado para la detección de error. El checksum convierte la data en segmento, su cabecera y también las direcciones de origen y de destino y el valor del protocolo desde la dirección IP. En una manera similar al UDP, si el receptor detecta que el datagrama se ha corrompido, este lo descarta. Pero, el TCP posee un mecanismo interno para el reenvio de data que no ha sido confirmado. Esto asegura que la data será entregada definitivamente.
- **Urgent Pointer (16 bits)**
  Este campo apunta al byte después de la urgent (urgente) data. Asi que, el receptor puede determinar cuanta data urgente está incluida en el mensaje.
- **Option Type (8 bits)**
  Actualmente una sola opción, la opción 2, está definida. Específica el Tamaño máximo de Segmento (Maximum Segment Size, MSS) para la conversación.
- **Option Length (8 bits)**
  Este campo específica la longitud de la opción. La opción 2 (MSS) es de 4 bytes de largo.
- **Maximum Segment Size (MSS) (16 bits)**
  El tipo de opción más común del TCP, específica el tamaño de segmento más largo que el TCP transmitirá al otro node. La información MSS es intercambiada durante el establecimiento de la sesión TCP.

**TCP Handshaking/Saludo**

Las seis opciones/flags de TCP involucran varios mensajes de control que son enviados para poder establecer la conexión y cerrarla. Estas opciones tienen los siguientes significados:

| | |
|---|---|
| SYN | Número de secuencia sincronizado para iniciar la conexión |
| ACK | Confirmación/Acknowledgment de la trama |
| RST | Reinicia la conexión |
| FIN | El emisor ha acabado de enviar la data |
| URG | Data entrante es Urgente |
| PSH | Pasar esta data a la aplicación lo más rápido posible |

El TCP establece una conexión vía un método conocido como saludo de tres via:

- El host requiriente envía un segmento con una opción SYN colocada en el campo CODE BITS conteniendo un número de secuencia inicial.
- El host receptor envía su número secuencial inicial de vuelta al host que hizo la petición, de nuevo en un segmento con la opción SYN establecida. También aprovecha la oportunidad de colocar un ACK del número de secuencia inicial en el segmento para el host que hizo la petición.
- El host requiriente ahora recibe el ACK del número de secuencia inicial asi que sabe que el otro lado está sincronizado con el. Todo lo que resta es que el host haciendo la petición confirme que ha recibido el número de secuencia inicial del host que recibe.

Normalmente, los segmentos son pasados por la saludo/handshake no contienen data. Note que la confirmación enviada son para el número de secuencia más uno. La confirmación apuntan al próximo byte de data que puede ser enviado.

Cualquier de los dos puntos pueden cerrar la conexión. Un segmento de FIN significa que la aplicación en la otra punta no estará enviando más data. Cada FIN deberá ser confirmado para que el que lo envió sepa que fué recibido.

Puede que piense que el ACK del primer FIN se aproveche y se envíe con el segundo. Pero, el segundo FIN sólo se generará una vez la aplicación remota cierre sus puntos de envío, asi significando que no enviará más data. Aunque un punto haya sometido un FIN, el otro punto puede continuar enviando data infinitamente. De hecho, algunas aplicaciones hacen uso de este tipo de conexiones media cerradas/half-closed. Solamente cuando ambos lados han sometido su confirmaciones a FIN es que la conexión será completamente cerrada.

Las siglas mss significan Maximum Segment Size (Tamaño máximo de Segmento) y es un limite en el tamaño máximo de las tramas de data que pueden ser enviadas al host. Esto es primordialmente útil si el host esta en una red con un MTU reducido y desea evitar demasiada fragmentación de la data.

La razón para tener cuatro señales de terminación pero sólo tres para establecer la conexión es que cada lado de la conexión debe ser apagado individualmente. El TCP es de comunicación full-duplex y se puede lograr half duplex ejecutando un half close, e.j., apagando un lado de la conexión.

**Opciones TCP**

Un paquete TCP puede tener opciones. Uno de los parámetros pasados por estas opciones es el Tamaño máximo de Segmento (Maximum Segment Size). Otra es la opción el Sello (Timestamp) la cual proveé una manera de registrar el tiempo que toma el viaja de ida y vuelta de la data TCP.

**Control de Flujo TCP**

La conversación entre el emisor y el receptor siempre incluyen una indicación de cuantos bytes de data pueden ser recibida a la vez sin riesgo de perdida, esto es llamado una ventana/window. El tamaño de la ventana es el número máximo de bytes que pueden ser enviados sin recibir una confirmación ACK. Cada segmento TCP contiene un valor de la ventana actual, en bytes, de la otra punta de la conexión.

Cuando la data es enviada, esto tiene el efecto de cerrar la ventana, en efecto, reduciendo el número de bytes que puede ser enviado actualmente. Cuando el otro lado del conexión recibe data, enviará una confirmación del número de bytes que ha recibido. Cuando se recibe la confirmación, esta tiene el efecto de abrir la ventana, en efecto, incrementar el número de bytes que pueden en la actualidad ser enviado. Este tipo de control de flujo es conocido como sliding window ya que la ventana se desliza sobre los bytes enviados.

El emisor puede calcular la ventana actualmente disponible del receptor del tamaño anunciado y del número de bytes enviado pero no confirmado. Cuando esta se convierte muy pequeña, el emisor no transmitirá hasta que el receptor anuncie que más data puede ser recibida.

Por ejemplo, si el receptor ha estado anunciando una ventana de 4,096 bytes en el segmento de su cabecera y el emisor ha enviado 2,000 bytes sin haber recibido una confirmación, la ventana restante fuese de 2,096 bytes. Si el emisor fuese a enviar otro de 2,096 bytes, el cual también es confirmado, este tendrá que restringirse de envíos posteriores hasta que no se reciba una confirmación. Si este entonces recibe una confirmación de los primeros 800 bytes enviados, entonces este podrá enviar otros 800 bytes.

El receptor almacenará los segmentos sin importar el orden en que fueron recibidos. Si el segmento llega fuera de orden o con faltante, el receptor confirmará el número de secuencia más grande, en efecto, el último punto en el flujo de data que este ha recibido la totalidad de los bytes. Este entonces esperará que el emisor de la señal de tiempo vencido (time out) y reenviará cualquier segmento que cree sea perdido.

Si se recibe un segmento duplicado, será descartado. Si se recibe un segmento fuera de orden, el último confirmado será reenviado. Asi que las confirmaciones duplicadas pueden darle al emisor una indicación de cual fué el primer segmento perdido.

**SYN Cookies**

Los SYN cookies son un mecanismo importante de seguridad en contra de ataques TCP SYN efectuados sobre un sistema. En la eventualidad de un ataque, SYN cookies vencen el ataque conservando memoria y no permitiendo que multiple peticiones de paquetes agoten la memoria. En esta sección discutimos como un desbordamiento (flooding) de TCP SYN funciona y como los SYN cookies puede ser usado para evitar caídas del sistemas a traves de ataques SYN.

**SYN Flooding**

Es muy posible que un servidor TCP/IP Web se tornará lento y hasta puede fracasar completamente si demasiadas peticiones de conexión se reciben simultáneamente. Este problema, bajo circunstancias legitimas, puede ser un problema de conexiones lenta o de un servidor que no puede procesar la alta demanda de paquetes TCP de conexión. Muy común es que personas (crackers) intencionalmente queriendo causar daños ataquen un servidor Web con un simple proceso conocido como SYN flooding, hoy día más comúnmente llamado Denial of Service attack o DoS.

Los ataques SYN flooding se aprovechan de la secuencia de eventos requeridas para establecer una conexión TCP. Normalmente, cuando una máquina se conecta a un servidor, esta envía un paquete TCP conocido como un paquete SYN (synchronization), cual el servidor Web recibe y responde enviandole un paquete de confirmación (ACK). Una ves la máquina conectandose recibe el paquete ACK, esta envía otro paquete TCP

con la opción ACK establecida, asi completando el proceso de conexión. Este proceso es a veces referido como un saludo de tres vía. Pero, cuando estamos bajo un ataque SYN flood, un gran número (posiblemente millones) de paquetes SYN son enviados, todos reclamando haber provenido desde equipos no existentes, a traves de creando fantasmas (spoofing) de direcciones de redes. El servidor trata de completar el saludo de tres vía, pero ya que la petición es de un sistema no existente, no pasa nada y el servidor deja el intento de hacer la conexión.

Un atacante usando un SYN flood puede fácilmente sobrecargar la memoria ya que cada petición de conexión requiriendo información del servidor afecta nuestro sistema. Los servidores Web son limitados en el número de paquetes SYN que ellos pueden recibir por puerto. Cuando paquetes no procesados empiezan a sobrecargar un puerto, los siguientes paquetes que arriban son descartados. Durante un ataque SYN flood, estos paquetes descartados con mucha seguridad que también eran parte de los paquetes SYN intentando conectarse o peor aún, si es un sistema intentando conectarse.

**La Defensa SYN Cookie**

Una defensa incluida en el kernel de Linux desde la versión 2.0.30 en adelante es encontrado del desborde SYN flooding es usar los SYN cookies (también conocido como los TCP SYN cookies). Los Cookies pueden diferenciar entre las conexiones legitimas y las ilegitimas sin usar muchos recursos de memoria. Los SYN cookies permiten que sistemas legitimo se conecten durante un ataque de SYN flood que normalmente fueran rechazados. Los SYN cookies trabajan codificando toda la información acerca de una conexión en el paquete ACK que es retornado al sistema que se conecta después de que el paquete SYN inicial es recibido por el servidor. Si la conexión es un actualmente un emisor y no desde una dirección IP falsa, el SYN cookie retornará el tercer paquete TCP que contiene toda la información necesaria sobre el sistema que se conecta. Esto significa que nada tiene que ser almacenado en la memoria que por lo normal bloquearía el servidor durante un desbordamiento SYN.

Para habilitar los SYN cookies en GNU/Linux, ejecute los siguientes comandos (como el root):

```
# echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

para deshabilitarlo, sólo repita la sentencia anterior pero asigne un cero 0 en vez de un valor de 1;

```
# echo 0 > /proc/sys/net/ipv4/tcp_syncookies
```

**Puertos y Sockets**

Necesitamos un método para canalizar los datos que fluyen desde y hasta a un proceso en particular de una aplicación. Este meto de canalizar datos es conocido como puertos y sockets. Un socket es un par de pila de datos para el proceso de una aplicación, uno en el punto de enviar o input y uno en el punto de recepción o el output. Un puerto es la dirección de un socket, igual al número de la extensión telefónica es la dirección de esa extensión.

Un Sockets por general es implementado como un área de memoria dentro del kernel de un sistema operativo. La data entrante es es destinada vía la dirección del puerto de un socket y por lo tanto directamente dentro del proceso de una aplicación.

El protocolo de transporte usa el concepto de un puerto para representar un punto d comunicación. Los puertos son asignados números, representados por un valor de 16-bit. Esto significa que ellos pueden ser hasta 64 KB de puertos individuales. El TCP y UDP cada uno tiene su propio conjunto de puertos.

Los puertos usados usados por TCP/IP son parecidos a los números de extension de un sistema telefónicos. Si deseamos contactar a un amigo via teléfono debes primero tener su número telefónico. Esto es el equi-

valente a usar una dirección IP para contactar un host. Una vez y lo has contactado, puede que necesite pedir una extensión particular para poder hablar con su amigo. Este es el equivalente al número de puerto.

La combinación de dirección IP y el número´mero de puerto es a menudo mejor conocido como un socket. Al combinarlo con el número de protocolo seleccionado, ya sea TCP o UDP, esto provee una dirección única de u proceso en un red basada en TCP/IP. Esto en en efecto es, un par de colas para la data de la aplicación, una de entrada y otra de salida.

El nombre del socket también es referido como un API (Application Programming Interfase), usado frecuentemente para el desarrollo de aplicaciones TCP/IP. La interfase de programación de sockets fué desarrollada en la Universidad de California Berkley cuando ellos implementaban TCP /IP en UNiX. Por esto a veces es referido como los sockets de Berkeley.

La interfase de sockets soporta toda la funcionalidad que un programador necesita para implementar un cliente o un servidor. Esta es una interfase por defecto en UNiX y ahora es común en otros ambientes que soportan sockets TCP/IP.

**Número de Puertos**

Los servidores escuchan por peticiones de clientes en puertos específicos, como esta definido en el RFC 1700. Los puertos números 1 al 1,023 son manejados por IANA. Teniendo los procesos de servidor escuchar por comunicaciones en u punto en particular es como los servicios son anunciados. Es normal tener los procesos del servidor escuchar en el todos los sistemas que proveen el servicio. Este es a menudo llamado el puerto de reencuentro. Por ejemplo, para contactar el servicio de telnet en un sistema, tendría que comunicarse usando TCP, con el puerto número 23 en ese sistema.

Los números de puerto entre 0 y 1,023 son reservados para servicios bien conocidos y solo pueden ser accesados por root. Estos puertos también son referidos como puertos privilegiados o o servicios bien conocidos. La mayoría de los servicios que se ejecutan en el Internet utilizan estos puertos bien conocidos. Esto es para que estos servicios siempre puedan ser contactados en el mismo puerto sin importar la dirección IP del sistema. Los usuarios que no son root pueden usar los número s de puertos que son de número más alto que el 1,024. Por eso es que puede ser vea el servidor HTTP ejecutándose en el puerto 8,000 o el 8080- ellos son ejecutados por otros usuarios que no son root ni otros administradores del sistemas.

Los servicios bien conocidos son definidos por una organización central de nombre IANA. Aunque los número de puertos TCP y UDP son independiente, Típicamente son asignados el mismo número bien conocido para ambos el TCP y el UDP.

Los procesos cliente requieren diferentes puntos para la comunicación, al igual que los servidores también. Eso si, no es importante que un cliente use siempre el mismo número de puerto. Cualquier número de puerto puede ser usado siempre y cuando el número de puerto usado es consistente durante una conversación con el servidor. Para permitirlo, TCP y UDP pueden dinámicamente seleccionar un puerto que no este en uso para (por lo general con números muy alto) darle a un cliente que lo requiere. Estos puertos son conocidos como números de puertos efímeros.

Los servidores escuchan por peticiones de conexiones en estos puertos específicos porque los clientes conocen estos puertos, y porque los puertos están estandarizado.

**Números de Puertos Importantes**

Algunos números de puertos han sido reservados para servicios comunes servidos en todo el Internet. Los listamos en la siguiente tabla.

| Puerto | Servicio |
|---|---|
| 7 | Echo |
| 13 | Date + Time |
| 20&21 | FTP |
| 22 | SSH |
| 23 | Telnet |
| 25 | Mail (SNTP) |
| 57 | Domain Name System (DNS) |
| 80 | www (HTTP) |
| 110 | Mail (POP3) |
| 111 | Portmapper (RPC) |
| 119 | News(NNTP) |
| 123 | Network Time Protocol (NTP) |
| 137 | NetBIOS name service (WINS) |
| 139 | NetBIOS session servile |
| 143 | Internet Message Access Protocol (IMAP2) |
| 161,162 | Simple Network Management Protocol (SNMP) |
| 389 | lightweight Directory Access Protocol (LDAP) |
| 443 | Secure Sockets Layer (SSL) |

Usted puede ver que números de puerto están disponible para brindar servicios en su sistema si damos lectura al archivo /etc/services. Este archivo contiene el mapeo de los nombres de servicios a los números de puertos. Algunos programas nos permiten usar el nombre del servicio en vez del número de puerto. Por ejemplo, ambos de los siguientes comandos abriran una conexión al servidor Web en la máquina de nombre web.abiertos.org:

```
telnet   web.abiertos.org 80
telnet  web.abiertos.org  www
```

### Pasar Data a las Aplicaciones

Cuando un datagrama IP es recibido por un host, este puede determinar cual protocolo de transporte este debe ser entregado examinando el valor en el campo protocolo del cabezal IP. Cada uno de los protocolos de transporte, TCP y UDP, soportan una amplia gama de servicios al nivel de aplicación. Es perfectamente posible, al punto hasta generalizado, tener más de un servicio activo a la vez. Asi que debemos tener una manera de identificar comunicaciones individuales en ambos puntos desde dentro del software del protocolo de transporte.

Como ya hemos visto, ambos cabezales del TCP y UDP contienen dos números de puertos, uno de destino y otro de origen. El número de puerto de destino determina a cual aplicación la data se le pasa.

Ahora un programador de aplicaciones solo tiene que preocuparse de el flujo de data que el esta usando y puede dejar que el TCP maneje la transferencia confiable, de flujo controlado. Esto simplifica enormemente la tarea de escribir aplicaciones.

### El Protocolo HTTP

El protocolo HTTP (Hypertext Transfer Protocol) esta definido en el RFC 1945 como el del World Wide Web. El tráfico WWW es pasado por el puerto 80, cual es el HTTP, asignando por el RFC 2086 en su versión 1.1. El SSl (Secure Sockets Layer) usa el puerto 443, conocido como el HTTPS.

Es importante denotar que el HTTP 1.0 requiere conexiones por separado para cada descarga de archivo o imagen. El HTTP 1.1 si permite que las conexiones se mantengan abiertas.

Recientemente desarrollado en el año 1992 por el científico Británico Tim Berners-Lee, el HTTP ha evolucionado para proveer un mecanismo estandar por el cual documentos pueden ser publicados y presentados en línea.

El protocolo HTTP es usado para transferir páginas WWW. Las páginas son escritas en el lenguaje de marcado de hipertexto HTML (Hypertext Markup Language). El HTML permite la especificación de diagramación, fuentes, colores y inclusión de gráficas en una página WWW. El HTML también define vínculos, los cuáles pueden ser usados para navegar de página en página a traves de efectuar clicks.

La aplicación que provee el servicio de vaginas y documentos WWW es llamada un servidor de páginas Web. El Apache (www.apache.org) es el servidor de páginas web más popular en uso en el Internet con una distribución de uso de más de un 70% y creciendo. La aplicacion cliente que utiliza los servicios WWW es llamado un navegador/browser, y presenta el material gráficamente y con el formato deseado en pantalla al usuario.

El SSL proveé un flujo de data encriptado que es utilizado para proveer comunicaciones encriptadas y seguras en el puerto 443 y el protocolo seguro HTTPS (Secure HTTP). Los sitios web que ofrecen servidores web con el SSl habilitado sobre el puerto 443 tienen URLs que se digitan con “https://dirección.dominio”.

**Servidores GNU/Linux**

GNU/Linux ofrece una gran variedad de servidores para los diferentes protocolos de alto nivel. Estos son identificados por el nombre común del protocolo más una “d”, e.j., telnetd y sshd. Los servidores también son conocidos como daemons en el léxico de GNU/Linux.

Los servidores o daemons TCP tienden a ser concurrentes; lo que significa que ellos esperan por la petición de un cliente, inician el nuevo proceso/tarea/hilo para manejar el cliente, luego retornan a esperar por la próxima petición.

En implementaciones anteriores de TCP/IP en GNU/Linux, un servidor tenía que estar ejecutándose por cada servicio que se ofrecía en su host. Pero, la mayoría de las veces, mucho de estos servers estarían inactivos, esperando por clientes. Lo cual significa que muchos recursos del sistema están siendo desperdiciados. Implementaciones más recientes usan una tecnología llamada los super server, su nombre en GNU/Linux es inetd, el cual espera por conexiones en todos los puertos y entonces enciende/spawns el servidor de la aplicación cuando la petición llega. Lo cual significa que sólo el super server se encuentra ejecutándose a menos que alguien este actualmente usando el servicio. El super server es frecuentemente usado y es configurado vía el comando /etc/inetd.conf. En GNU/Linux, el TCP/IP es altamente optimizado, asi resultando en implementaciones mucho más rápido de TCP/IP.

**Ejercicio 4-1: Seguridad de la Red de la Compañía X**

No se proveen soluciones a este ejercicio. La red está ya en un estado muy avanzado. Usted es el administrador de la red. Aqui se le provee las necesidades de diferentes departamentos de la Compañía X, y usted está en el deber de proveer estas necesidades pero siempre tomando en cuenta la necesidades de seguridad.

Es necesario que piense en todo el disenso de la Compañía X. Trate de responder las siguientes preguntas:

1.- ¿Cuáles medidas de seguridad de la red puede usted usar?

2.- La oficina principal no esta expuesta al Internet. ¿Qué tipos de defensas son apropiadas para estos siste-

mas?

3.- El departamento de Ingeniería necesita la máxima protección. ¿Qué sugiere usted aquí en este caso?

4.- El equipo de desarrollo necesita acceso rápido y fácil a recursos en línea pero posee datos sensibles que no pueden ser robados. ¿Puede usted pensar en una solución que satisfaga ambas de estas necesidades?

5.- También considere que otras sugerencias y medidas de seguridad puede usted tomar para asegurar que nuestra oficina no sea blanco de ataque.

## Ejercicio 4-2: Completar la Configuración de Red

No se proveen soluciones a este ejercicio. Deberá tener los permisos de superusuario para editar el archivo hosts. Si lo edita en vi y le informa que está en modo de solo lectura, solo tendría que al momento de guarda hacerlo con el comando :w! de vi para forzar vi a escribir el archivo.

Este es un ejemplo del archivo hosts:

```
# Mi archivo hosts
127.0.0.1       localhost
192.168.1.5     almacen
192.168.2.7     proxy
```

Ahora usted debe poder hacer ping a estos equipos solo usando el nombre y no el IP. Aseguro que usted puede hacer esto.

Pruebe haciendo ping a un sistema en otra red (use la dirección IP del sistema). ¿Qué mensaje usted ve?

Ahora asignele a uno o más de los hosts en su archivo hosts un alias. Revise que ahora pueda contactar el sistema usando:

- **Dirección IP**
- **Nombre de Host**
- **Alias**

### *Acceso/Login Remoto a Otros Sistemas*

Usted debe poder poder repetir su login remoto a otros sistemas, pero esta vez usando el nombre del sistema y no la dirección IP.

Use el comando telnet para accesar el sistema remoto:

**$ telnet nombre-sistema**

Este comando le presentará un prompt pidiendole un nombre de usuario y contraseña.

Una vez usted a ingresado a sistema remoto, escriba el siguiente comando:

**$ uname -n**

Este comando imprimirá el nombre de host del sistema que usted se encuentra, logueado remotamente.

### *Monitorear la Actividad de la Red*

El comando netstat es útil ya que nos da la habilidad de monitorear el estatus del software utilizando la red en el sistema. Para esta sección, usted debe disponer de por los menos dos ventas abiertas; una para ejecutar el comando netstat para examinar su salida y la otra para ejecutar comandos de red.

Escriba el siguiente comando:

**$ netstat -a --inet (netstat -af inet en algunos sistemas)**

Para desplegar el estado de los puntos TCP y UDP en su sistema.

Al examinar los punto del TCP en su sistema, usted debe ver que las mayorías de puntos están en un estado de LISTEN/Escuchando. Esto significa que no haya conexiones activas en este momento, pero que el servidor esta activo y escuchando.

Ahora desde la otra venta que abrimos, ingrese en un sistema remoto usando el comando telnet, como hizo anteriormente. Una vez ingresado al sistema, escriba el siguiente comando:

**$ netstat -a --inet**

Note que ahora su conexión ya esta indicada. La sesión de telnet mostrará un esta de conexión en el estado de de ESTABLISHED/establecida. Si otros usuarios se encuentran ingresados en su sistema, sus conexiones también estarán marcadas. Usted debe poder deducir cuáles sistemas tienen usuarios logueados en la actualidad en su sistema desde esta información.

Si usted escribe:

**$ netstat -ei**

Usted será presentado un resumen de la estadística de las interfases de redes conectadas e instaladas al equipo. Alternativamente, el comando netstat puede ser ejecutado con la opción -c para continuamente actualizar su salida (aunque esto tiende a ser difícil de leer, particularmente si la salida no es alimentada por tubería a un paginador o lector). Pruebe con la combinación de opción -ae para ver que conexiones son para que servicios.

Cada servicio basado en TCP o UDP en su sistema esta asociado con el número de un puerto. Los detalles de los servicios, sus protocolos de transporte y los números de puertos se almacenan en el archivo /etc/services.

Usando sólo la información en este archivo, llene el siguiente formulario de los protocolos de transporte y los números de puertos usados por los siguientes servicios:

| | | |
|---|---|---|
| **time:** | **----------** | **----------** |
| **echo:** | **----------** | **----------** |
| **tftp:** | **----------** | **----------** |
| **name server:** | **----------** | **----------** |
| **telnet:** | **----------** | **----------** |

**Ejecute de nuevo el comando netstat con la opción numérica y observe como esto se refleja con el número services:**

**$ netstat -an --inet**

Finalmente usted debe estar consciente como es que GNU/Linux controla los servicios principales. Esto se resumen en las siguientes secciones. Read the following information to raise your awareness.

***El inetd***

GNU/Linux, al igual que su antecesor UNiX, posee mecanismos de ahorro de memoria para proveer servicios vía el proceso de super-server. El mapeo de los puertos a los servidores tiene dos etapas:

- El archivo /etc/services mapea los número de puertos al nombre de los servicios.
- El archivo /etc/inetd.conf mapea los servicios a los nombres de los servidores.

El inetd monitorea todos los puertos en el archivo inetd.conf esperando por intentos a conexiones. Al recibir una en un puerto en particular, establece la asociación con el nombre de servicio asociado a este puerto en el archivo services. Ya establecido esto, busca en la información que recibió del archivo inetd.conf. Esto le dice varias cosas acerca de la conexión, la más importante siendo donde reside el servidor de ese servicio, entonces inetd iniciará el servidor y entregará la conexión al servidor. Luego retorna a escuchar en el puerto. Las conexiones individuales al mismo servicio o número de puerto son diferenciadas por la combinación de la dirección única de IP y el número de puerto.

El archivo de servicio es un mecanismo extremadamente flexible. Con simplemente cambiando la entra-

da relevante, es posible alterar el puerto que el servicio usará de ahí en lo adelante.

Este es un formato básico del archivo inetd.conf:

| Servicio | Endpoint | Protocolo | Wait | Usuario | Ruta | Argumentos |
|---|---|---|---|---|---|---|
| finger | stream | tcp | nowait | nobody | /usr/libexec/tcpd | fingerd-s |
| ftp | stream | tcp | nowait | root | /usr/libexec/tcpd | ftpd-l |
| login | stream | tcp | nowait | root | /usr/libexec/tcpd | rlogind |
| nntp | stream | tcp | nowait | usenet | /usr/libexec/tcpd | nntpd |
| ntalk | dgram | udp | wait | root | /usr/libexec/tcpd | ntalkd |
| shell | stream | tcp | nowait | root | /usr/libexec/tcpd | rshd |
| telnet | stream | tcp | nowait | root | /usr/libexec/tcpd | telnetd |
| uucpd | stream | tcp | nowait | root | /usr/libexec/tcpd | uucpd |
| comsat | dgram | udp | wait | root | /usr/libexec/tcpd | comsat |
| tftp | dgram | udp | wait | nobody | /usr/libexec/tcpd | tftpd /private/tftp |
| bootbootps | dgram | udp | wait | root | /usr/libexec/tcpd | bootpd |
| pop3 | stream | tcp | nowait | root | /usr/libexec/tcpd | /usr/local/libexec/popper |
| imap4 | stream | tcp | nowait | root | /usr/libexec/tcpd | /usr/local/libexec/imapd |

El super servidor inetd, /usr/shin/inetd, lee la información que describe los servicios que el ha de controlar desde el archivo de configuración /etc/inetd.conf. Las entradas en este archivo describen los servicios de la siguiente manera:

**Service** Nombre del servicio. Debe corresponder con una entrada en /etc/services y es usado para decirle a inetd cual puerto monitorear para este servicio.
**Endpoint** Define si el punto final del servicio va a operar las transferencias con flujos/stream o basados en datagramas.
**Ptcl** Este es el protocolo que es usado para el servicio.
**Wait** Define si múltiples clientes concurrentes son soportados (nowait) o no.
**User** El nombre de usuario bajo el cual se ejecuta el servicio.
**Pathname** La ruta del programa a ejecutarse.
**Arguments** Los argumentos (incluyendo el nombre del comando) pasados en el momento de e al programa servidor.

Para desactivar un servicio simplemente se elimina la entrada del archivo /etc/inetd.conf o se coloca (#) como un comentario al inicio de la línea que queremos anular. El archivo es leído una vez cuando el inetd se inicia al momento de boot, los cambios deben ser notificados a inetd. Esto se logra enviandole una señal de SIGHUP (señal no. 1), por ejemplo:

```
# ps  -aux  |  grep inetd
root  321    0.0     1.2      892      392     ?    S    May  13      0:22 inetd
#  kill   -HUP 321
```

## El Protocolo UDP

El TCP si es complejo; este debe proveer confiabilidad (confirmaciones/acknowledgments), flujo (saludo/handshaking), control de flujo (tamaño de ventana/windowing), soporte para interrumpir el flujo y más. Todo esto suma a la complejidad del TCP y subtraer del desempeño del protocolo. Para algunas aplicaciones, estos requerimientos no son necesarios. Estas aplicaciones necesitan una alternativa, UDP.

Los siguientes tópicos son discutidos en esta sección:

- Fundamentos de UDP
- Transferir Archivoa vía TFTP
- Números de Puertos UDP

## Fundamentos de UDP

El UDP puede ser usado cuando confiabilidad no es importante y cuando no es necesario el flujo de data, esto significa que no hace falta la numeración de las tramas.

## Formato de la Trama

El cabezal incluido en cada datagrama por el UDP es simple, consiste principalmente de información que permite que el UDP efectué su entrega de la data al proceso apropiado (los números de puerto).

El cabezal UDP contiene los siguientes campos.

- **Source Port (16 bits)**
  Campo que define el número de puerto del servicio que genera el mensaje en nodo de origen.
- **Destination Port (16 bits)**
  Este campo identifica el número de puerto del servicio en el nodo destino.
- **Message Length (16 bits)**
  Este campo identifica el número de bytes en el paquete UDP (incluye el cabezal UDP y la data). Esto es realmente redundante ya que el cabezal UDP es de longitud fija y el tamaño de la data UDP puede ser calculada tomando en cuenta solamente la longitud del datagrama IP. La longitud es un valor sin signo de 16-bit, dando un datagrama UDP máximo de 65,535 bytes, o 64 KB.

- **Checksum (16 bits)**
  Este campo es opcional. Un checksum de valor 0 implica que el checksum no fué corrompidos. El Checksum cubre toda la data en el datagrama UDP, su cabezal de 8-byte y también la dirección IP de origen y destino y el valor del protocolo de la dirección IP. Si el receptor detecta que el datagrama esta corrompido, simplemente lo descartas, al igual que el IP. Es el deber de las aplicaciones de más alto nivel usando el UDP efectuar su propia detección y corrección de error.
  Algunas aplicaciones deshabilitan el checksum del UDP para reducir el costo del uso de recursos. En una LAN, esto es razonable ya que la capa de Enlace por lo general detecta cualquier tipo de corrupción. Pero, al trabajar sobre el Internet (internetwork), el checksum debe siempre estar habilitado ya que no existe otra manera para el receptor revisar la integridad de la data.

El cabezal combinado con la data es encapsulada como la porción de data del datagrama de IP. El datagrama IP también contiene las direcciones del sistema de origen y destino y el campo del Protocolo IP contendrá un valor de 17, indicando que la data es un datagrama UDP.

### Cuestión de Confiabilidad

Claro está, este protocolo no es útil cuando es importante la confiabilidad. Aquí le presentamos una lista de servicios que por lo general dependen de una conexión confiable y debido a esto no son recomendable a través del UDP:

- FTP
- Mail
- Telnet
- WWW

### El UDP en la Practica

El UDP es un protocolo simple, que esencialmente provee una interfase de alto nivel a la facilidades del IP para transportar datagramas. La data es transmitida en datagramas sin garantía de entrega; si confiabilidad es importante, entonces es delegada al software de más alto nivel para que la provea.

El UDP coloca un bajo costo fijo en termino del control de información de protocolo en la data transmitida. Es muy a menudo considerado mucho más eficiente que su alternativa el TCP, y realmente en un ambiente LAN, por lo general es más que adecuado como un protocolo de transporte. Muchas aplicaciones de área local utilizan el UDP por esta razón. En redes de gran escala que involucran muchos gateas y enlaces de mayor área, la falta de un mecanismo confiable de entrega interno del protocolo se convierte en un problema y en este caso el TCP se convierte en una mejor elección.

Generalmente los usuarios accesarla red a través de un servicio del nivel de aplicación, y la elección entre el UDP y el TCP ya habrá sido efectuada por el programador de la aplicación.

En la manera que un servidor se comporta depende de que protocolo de transporte este usa. Los servidores que usan UDP por lo general serán iterativos. Lo que significa que ellos manejaran cada petición a medi-

da que lleguen y no atenderán otras peticiones hasta que no terminen con la actual. Esto cabe en el hecho que los datagramas UDP son entidades independiente; asi que ellos deben contener toda la data necesaria para identificar la petición.

**Transferencia de Archivos vía TFTP**

El TFTP (Trivial File Transfer Protocol) es asignado el puerto UDP número 69 definido en el RFC 783. El TFTP es mucho más pequeño y menos sofisticado que el FTP. Este usa el UDP como su medio de transport, lo cual simplifica considerablemente los programas clientes y servidores. También contiene un conjunto mucho más pequeño de comandos que solo se conciernen con la transferencia de archivos.

En una sesión básica empieza con una petición de escritura desde el cliente al puerto UDP 69 del servidor. El servidor entonces responde con una confirmación. Pero note que que esta confirmación y las conversaciones de ahi en lo adelante usaran un puerto en el servidor UDP diferente, el puerto es el 1390, el cual es un puerto efimero/ephemeral port. Las conversaciones continúan y concluyen en este puerto y en el puerto original del cliente.

¿Por qué es esto? Bien, el server TFTP desea liberar el puerto 69 lo más pronto posible. Bajo el TCP, tenemos un protocolo orientado a conexión con una conexión identificadas por ambos puntos de la conversación, como es, la dirección IP del cliente y el servidor, los puertos en cada punto y el protocolo (TCP). Asi pues, es perfectamente posible tener múltiples clientes conectarse al mismo puerto TCP del servidor porque la combinación de IP y puerto del cliente serán diferente para cada conexión.

Pero bajo el UDP, no tenemos una noción de conexiones. Es perfectamente posible que el servidor TFTP mantenga información de los IPs y puertos de los clientes y asi poder manejar multiple conexiones, pero todo esto tiene que ser llevado acabo en al nivel de las aplicaciones y requieren un esfuerzo adicional del programador. Además, lo más seguro que disminuiría la velocidad del servidor TFTP, y la intensión del TFTP es proveer un método simple y rápido de transferir archivos, y esto agrega una complejidad innecesaria. Asi pues, el el caso previo, el servidor TFTP resuelve este problema simplemente capturando un puerto efimero número 1390 y lo usa, y se va del puerto 69.

Si buscamos en GNU/Linux en el archivo /etc/inetd.conf, verá que el tftp esta marcado con “wait”; esto es que, las nuevas peticiones TFTP al puerto UDP 69 deberán esperar hasta que se libere para aceptar nuevas peticiones.

Cada transferencia empieza con un cliente efectuando una petición de lectura (RRQ) o de escritura (WRQ) al servidor. Una vez el servidor se ha asegurado que puede leer o escribir a un archivo dado, el host enviando la data envía el primer bloque. El receptor confirma el block antes de que otros bloques más sean enviados. Si no se reciben confirmaciones dentro de un tiempo específico, el emisor reenviará el bloque. El TFTP es un ejemplo de un protocolo pare-y-espere (stop-and-wait).

Debido a su pequeño tamaño y simplicidad, el TFTP es adecuado para la transferencia de archivos de inicialización durante procedimientos de arranque/boot. Es posible codificar un cliente TFTP para que quepa en un ROM en máquinas sin discos/diskless y asi formar parte del bootstrap (conjunto de archivos encargado de dar inicio al equipo) para esa máquina.

El TFTP no autoriza los usuarios como lo hace el FTP. Esto es un potencial punto débil de seguridad, ya que cualquiera puede tener acceso a copiar archivos desde y a nuestro sistema.

**Números de Puertos UDP**

Los servidores escuchan o atienden puertos específicos mientras esperan por las peticiones de los clientes, como es ilustrado en el RFC 1700. Los números de puertos del 1 al 1,023 son administrados por IANA, y son llamados los puertos bien conocidos o well known services. El rango entre el 1,024 al 5,000 son reservados para los procesos de los clientes, clientes son las aplicaciones. Los puertos por encima del 5,000 son conocidos como los NO bien conocidos (Non-Well known). Son utilizados para las aplicaciones de redes desarrolladas internamente en una empresa.

UDP tiene su propio conjunto de número de puertos; estos se ajustan exactamente igual que el esquema de los números de puertos TCP. Cuando el IANA asigna un número de puerto a un servicio, le asigna este número de puerto a ambos el TCP y el UDP, aunque el servicio no los use a ambos.

**RESUMEN**

En este capítulo, estudiamos el ICMP y los protocolos de la capa de transporte, UDP y TCP. Los siguiente es un lista de los puntos más importantes a recordar:

- ICMP es un protocolo simple usado para ayudar a mantener el Internet.
- ICMP es poderoso y puede rápidamente detectar donde empiezan y ocurren problemas de red.
- UDP es usado por servicios sin conexiones donde confiabilidad no es importante o necesaria.
- TCP es usado para servicios orientados a conexión y donde se necesita cierto grado de confiabilidad.
- TCP es más complejo y conlleva ciertos costos fijos en sus servicios de transporte.

**PREGUNTAS POST-EXAMEN**

Respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué ventajas tiene el TCP sobre el UDP? ¿Cual es usado más a menudo?

2. Describa el Cabezal IP.

# CAPITULO 5

*REDES DEL SISTEMA OPERATIVO GNU/LINUX*

# RESOLUCIÓN DE AVERIAS Y ADMINISTRACIÓN DE RED


| TÓPICOS PRINCIPALES | No. |
|---|---|
| Objetivos | 139 |
| Preguntas Pre-Examen | 139 |
| Introducción | 140 |
| Fundamentos de Localización de Averias | 145 |
| Resolución de Averias TCP/IP | 149 |
| Escenarios de resolución de Averias | 152 |
| Monitoreo de Rendimiento | 158 |
| Administración de Red y SNMP | 161 |
| Resumen | 164 |
| Pregunta Post-Examen | 165 |


## OBJETIVOS

Al finalizar este capítulo, usted estara preparado para efectuar las siguientes tareas:

- Dar solución a problemas de averias de TCP/IP
- Describir tecnicas comunes en la solución de problemas de redes.
- Comparar y contrasta las herramientas gráficas de configuración de redes: YaSt2, Netconf, NetConfig, y Turbonetconf,
- Describir la herramienta de administración de la red Netperf, particularmente los servicios que el programa proporciona.

## PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuáles son los tres pasos dado para comprobar para saber si hay conectividad?

2. ¿Durante el uso de un ftp, como podía corrumpirse un archivo?

3. ¿Qué efectua el netstat?

4. ¿Cuáles son los tres componentes de un sistema moderno de la dirección de la red?

# INTRODUCCIÓN

Muchas herramientas están disponibles para las redes en la localizacion TCP/IP. Este capítulo introducirá los archivos, las utilidades, y los comandos más importantes para localizar averías y discutirá cómo utilizarlos. Porque éstos tomaron se utilizan comúnmente con las redes de TCP/IP, muchas de ellas pueden ser modificadas y ser utilizadas a través de cualquier plataforma que funciona la habitación de TCP/IP.

La resolución de problemas del funcionamiento de una red de TCP/IP en el Distribución anbiente de sistema de hoy requiere analizar el efecto de la actividad del cliente y del servidor en sus sistemas, la red, y sus usos client/server.

Típicamente, el usuario del extremo es el primer para notar si una red se está realizando mal.

Este capítulo identifica los elementos que afectan el funcionamiento de las redes de TCP/IP. También cubriremos cómo los parámetros se asociaron a sus sistemas, la red, y sus usos client/server se pueden optimizar para reducir al mínimo problemas de la red. Los ejercicios de la resolución de problemas a través del capítulo asociarán los conceptos dominantes a problemas de funcionamiento reales de la red.

Mientras que i: no es posible cubrir una tesis profundizada 011 la resolución de problemas, este capítulo le señala en la dirección derecha.Con estas extremidades, usted puede buscar los libros, RFCs, cuartos de la charla, y newsgroup de ayudar a encontrar el ro de las soluciones sus problemas del establecimiento de una red. Este capítulo también da extremidades de la dirección de la red.

# FUNDAMENTOS DE RESOLUCION DE PROBLEMAS

En esta sección, hecharemos una ojeada centi'metro-sory cómo TCP/IP que localiza averías se logra. examinamos los varios métodos para localizar averías y las herramientas y los procedimientos que están disponibles para asistirle.

En esta sección discutimos los siguientes temas:

- Principios generales para localizar averías
- Caja de herramientas De Troubleshooter.
- Problemas Físicos más Comunes
- Problemas Lógicos más Comunes
- Problemas Especisicacion de Protocolos

## Principios Generales para Localizar Averías

Hay muchas metodologías para localizar averías. Ni continuar una discusión exhaustiva de metodologías; algo, hecharemos una ojeada una estructura básica de la metodología para localizar averías, aplicable en la mayoría de las situaciones.

***Reconozca los síntomas***

Muy a menudo, uno no puede solucionar el problema porque uno no lo ha identificarlo correctamente.

***Investigue el problema***

Debe siempre asegurarse de identificar donde y cuando el problema comenzó tan bien como cuántos sistemas pueden ser afectados semejantemente.

***Adquiera Información***

Este paso puede ser una consecuencia del paso anterior. Se cerciora de siempre usted tener toda la infor-

mación disponible de las fuentes, de la gente, y de los sistemas implicados.

***Pruébelo***

Deje su ayuda del diagnóstico de usted y sea seguro interpretando sus resultados exactamente.

***No salte a las Conclusiones***

A menudo el primer impulso que usted tiene puede curar el síntoma pero no el problema subyacente. Asegúrese siempre de usted haber cavado profundamente bastante para aislar la fuente del problema, la curación no justa el síntoma.

***Encuentre la Avería***

Este paso se podía también llamar encontrar la causa a los efectos descubiertos.

***Lista de lo qué Podría Ser***

No subestime la reunión de reflexión y la suerte en este paso. Ayuda a menudo a conseguir varios diversos pares de ojos y la habilidad fija implicado en identificar causas posibles (el tliat está, trabajo collabotarive de los esfuerzos mejor).

***Elimine los Problemas Uno Por Uno***

Cuando usted no sabe cuál es es el problema, pero usted sabe lo que podría ser, comienze a eliminar las menos probables. Usted puede encontrar a menudo que las causas son aisladas mejor con el proceso de la eliminación que intentando corregir lo que no es el problema. Recuerde que mientras más causas no posibles son eliminadas, menos el trabajo que tendrá que efectuar para corregir el problema.

***Corrija la Avería***

Ésta es normalmente la parte fácil. Cerciórese de que usted cure la enfermedad, no apenas el síntoma.

***Analice la Falla***

Ahora que no hay presión puesto que pudo corregir el problema, haga las preguntas importantes: ¿Se repetirá? ¿Es un sintoma de otros problemas? ¿Qué contribuyó a este suceso? ¿Cómo prevengo esto en el futuro?

## Caja de Herramientas para Corregir Averias

Recuerde que mientras más herramientas de corregir fallas posea mejor serán sus chances de éxito. Herramientas para corregir fallas de ambas indole de software y de hardware, asi como la preparación fundamentada y la experiencia de quien le pueda ser de asistencia, pueden hacer la resolución de problemas más rápida y más fácil.

***Herramientas De Software***

Entre algunas de las herramientas de software útiles para asistir en la resolución de problemas TCP/IP se pueden incluir:

- ping
- netstat
- traceroute
- nslookup
- ifconfig
- lsmod
- route
- ps

Los ficheros del log pueden también ser especialmente útiles. Revise siempre y cuando sea posible los ficheros de diario, por existencia de los mensajes de error. Incluso los mensajes no relacionados con el problema particular que usted está tratando de solucionar pueden resultar provechosos asi como pueden asistir-

le a no perseguir pistas erroneas y evitar asi perder un tiempo valioso en aíslar la causa principal del problema. Siempre y cuando pueda habilite el servicio de log de mensajes, particularmente los críticos.

***Herramientas De Hardware***

Los analizadores de red pueden ser particularmente útiles en el diagnóstico de problemas de red, y en algunos casos, pueden hastas recomendar curso de acción para remediar los problemas. Los analizadores de red disponibles pueden ser basados en hardware o software y dedicado o nondedicados. Los fragmentos de paquete capturados (y por lo general interpretados) por los analizadores de red son el mejor método para determinarse qué realmente está sucediendo en su red. Al hacer su investigación siempre recuerde que los administradores de sistema, los usuario e incluso las máquinas y los softwares pueden mentir, pero el cable nunca mintirá.

***Otras Herramientas***

No se olvide de que su herramienta más útil es su cerebro y los otros cerebros que usted puede integrar a la solución del problema.

***Problemas Físicos Comunes***

Problemas fisíos que involucran conectividad y enrutamiento.

***Conectividad***

Pongale mucha atención a las cosas más simples y obvias. Muy a menudo la resolución de un problema de alto nivel puede ser simplificado observando deficiencias obvias en el sistema.

Primero, ¿Están todos los cables conectados correctamente? ¿se encuentran conectados el teclado y el ratón? ¿está el cable de red en el lugar correcto y en perfecto estado? Se han dedicado muchas horas inecesarias localizando el problema de red. Esto ocurre muchas veces, cuando sólo era un cable desconectado, por accidente u otras fallas simples. Siempre haga las cosas simples primero y busque lo obvio y no lo complejo inicialmente al empezar a localizar averías.

***Enrutamiento***

Si usted se encuentra en una red enrutada, lo que es la generalidad de hoy en día, la resolución de problemas en su caso llega a ser un poco más compleja para usted determinar en qué segmento o subred está la avería. La detección de esto puede implicar muchos pasos, incluyendo (pero no limitado a):

- ¿Muestra el ping pérdida de paquetes?
  - Una conexión lenta puede ser la causa de éstas pérdidas.
  - O no trabajar del todo.
- Si sus manejadores de red se cargan como módulos en el kernel, puede ser que necesite utilizar el comando lsmod para asegurarse de que el módulo apropiado este cargado en el kernel.

Luego, revise las tablas de enrutamiento para verificar que estén válidas y correctas. Utilice el comando route para comprobar sus tablas de enrutamiento IP.

Usted puede utilizar el comando netstat para visualizar el estado de las conexiones de red actuales y con la opción -a para imprimir todos los sockets, incluyen los que se encuentra escuchando por peticiones.

Asegurse de detener conexiones en ambos puntos. Los paquetes pueden fluir desde un punto pero no desde el otro, causando la confusión y enmascarándo donde sucede el problema.

## Problemas Lógicos Comunes

Los problemas lógicos conciernen con archivos de configuración, un proceso del servidor, la pila del TCP/IP, servicio de nombre y archivos de diario.

***Archivos de Configuración***

¿Le parecen aceptables? Sea especialmente cuidadoso de los cambios recientes, especialmente ésos que coinciden sobre preguntas relacionadas con el problema en cuestión.

***Procesos del Servidor***

Debe preguntarse ¿Se encuentra el servidor en cuestión ejecutándose? Para determinar esto, intente conectar se con él con una sesión de telnet. Utilice el comando ps -ax para visualizar los procesos actualmente ejecutándose.

¿Es el proceso en ejecusión el proceso correcto (es decir, está el servidor en ejecusión corriendo el proceso que se supone?

No se olvide asegurarse que todos los sistemas no tengan ningún servicio extraño o innecesario ejecutándose. Éstos podrían estar interfiriendo con los servicios necesarios, y no todos los sistemas tienen todos los servicios habilitados.

***La Pila TCP/IP***

Primero, asegúrese que la interfaz TCP/IP está habilitada, ejecutándose y configurada correctamente. Utilice el comando ifconfig para comprobar y cerciorarse de que la interfaz está activa y con los ajustes de IP y la nermask correcta. Luego, intente de ejecutar un ping al localhost. Si esta prueba falla, usted tiene un problema de configuración.

¿Es su dirección IP correcta? Si la respuesta es no, puede ser que tenga un conflicto de dirección IP con otra máquina o que tenga una dirección inválida o corrompida.

¿Está la mascara de red (netmask) configurada correctamente? Si la respuesta es no, puede ser que se encuentre en la subred incorrecta. Esto puede conducir a muchos errores, especialmente al intentar alcanzar un host patticular.

***Servicio de Nombre***

¿Puede usted actualmente resolver nombres de host? ¿Puede usted resolver el nombre de host que usted está utilizando? Si la respuesta es no, intente usando la dirección IP.

***Ficheros de Diario y Estadística***

¿Ha generado un registro de error en los archivos log cualquier parte de la pila TCP/IP? Revise de log en el directorio /var/log/messages por cualquier reportado. Preste mucha atención en particular a estadísticas desequilibrada, e.j:, varios paquetes ICMP recibidos, ningunos enviados.

***Problemas Específicos de Protocolos***

Cada protocolo tiene diversos medios de la comunicación, así que cada uno tendrá un modo particular de falla. Es bueno saber algunos de los problemas comunes que pueden ocurrir en cada uno.

***Problemas del Ftp***

A continuación le presentamos una lista de algunos problemas comunes del ftp y cómo diagnosticarlos:

- **Tiempo de Conexión Agotado**

Recuerde que el ftp utiliza un contador de tiempo para detectar conexiones perdidas. Fijese cuánto tiempo toma un ping. Si se toma mucho tiempo, esta es la causa probable de la culpa o el factor que contribuye a esta falla.

- **Conexión Rechazada**

  ¿Soporta el host FTP? Utilice el nslookup para verificar.
- **Archivos Corrompidos**

  ¿UHa utilizado usted el modo correcto de transferencia?
- **FTP Anónimo**

  Browsers genéricos, como son por ejemplo el Mozilla, Firefox, pueden que necesiten que la dirección de correo electrónico sea configurada. ¿Ha seguido usted todos los pasos para configurar su servidor?

***DNS***

Errores de DNS son comunes en muchas redes. En esta sección, enumeramos algunos de los problemas relacionados con los DNS y sus posibles remedios para corregirlos.

¿Ha configurado usted correctamente su resolución de nombres? Si es así, usted debe poder entrar en contacto con su servidor de nombres. Intente hacerle ping a su servidor de nombres para comprobarlo.

Ponga a prueba es espacio de nombre de dominio usando los siguientes comandos y utilidades:

- **nslookup**

```
$ nslookup
> Server
Default Server:    codigolibre.org
Address:                66.17.131.10
> set q=any
> redhat.com
Server:         10.0.0.1
Address:        10.0.0.1#53

Non-authoritative answer:
redhat.com      nameserver = ns2.redhat.com.
redhat.com      nameserver = ns3.redhat.com.
redhat.com      nameserver = ns1.redhat.com.

Authoritative answer can be found from:
```

- **Domain Internet Gropet (DIG)**

El comando dig (Domain informatioI Groper) constituye una herramienta para realizar consultas de diverso tipo a un servidor de DNS. Este muestra las respuestas recibidas de acuerdo a su solicitud. Es muy útil para detectar problemas en la configuración de los servidores de DNS debido a su flexibilidad, facilidad de uso y claridad en su salida.

Aunque normalmente las consultas que permite dig se definen en la línea de comando también se puede hacer en un fichero y pasárselo como argumento (opción -f). En el caso de que no se indique el servidor a consultar se asumirán los especificados en /etc/resolv.conf. Cuando no se añade ninguna opción o argumento en la línea de comando se consultan los servidores de nombres del dominio raíz (NS query).

La forma básica de invocar a dig es:

dig <@servidor> <nombre> [tipo]

donde:

- @servidor - es el nombre o la dirección IP del servidor a consultar.
- nombre - es el nombre de dominio del record por el cual se quiere preguntar.
- tipo - es el tipo del record por el que se consulta (ANY, NS, SOA, MX, etc.). De no indicarse se asumirá A.

Sintaxis: dig [@servidor] [opciones] [nombre] [tipo] [clase] [opciones de consulta]

Algunas opciones:

- -h : muestra la ayuda del comando.
- -x : hace consultas inversas, o sea, a partir de las direcciones IP determina nombres de dominio.
- -f <filename> : toma las consultas a partir de un fichero. Estas se definen una por línea y con la misma sintaxis que en la línea de comand

**Ejemplos:**

```
$ dig @alma maildi +search
; <<>> DiG 9.1.1 <<>> @alma maildi +search
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6750
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;maildi.disaic.cu.              IN      A
;; ANSWER SECTION:
maildi.disaic.cu.       86400   IN      A       192.168.100.3
;; AUTHORITY SECTION:
disaic.cu.              86400   IN      NS      alma.disaic.cu.
;; ADDITIONAL SECTION:
alma.disaic.cu.         86400   IN      A       192.168.100.2

;; Query time: 75 msec
;; SERVER: 192.168.100.2#53(alma)
;; WHEN: Sun Apr 29 17:25:05 2001
;; MSG SIZE  rcvd: 85

$ dig @192.168.200.3 shine.disaic.cu +short
; <<>> DiG 9.1.1 <<>> @192.168.200.3 shine.disaic.cu +short
;; global options:  printcmd
192.168.200.2

$ dig disaic.cu NS
...
;; ANSWER SECTION:
disaic.cu.              86400   IN      NS      alma.disaic.cu.
disaic.cu.              86400   IN      NS      odin.disaic.cu.
...

$ dig -x 192.168.200.4
...
;; QUESTION SECTION:
;2.100.168.192.in-addr.arpa.    IN      PTR
;; ANSWER SECTION:
4.200.168.192.in-addr.arpa. 86400 IN    PTR     sion.disaic.cu.
...
```

Compruebe sus archivos maestros. Puede ser que le falten algunos puntos al final de algunos FQDNs. Volque los datos, que puede ser que revisando encuentre un nombre de dominio u opción mal deletreada.

***Errores del SMTP***

Observe siempre la cabecera de retorno para ver donde el mensaje falló y adonde fué. Revise los host To: y Cc: existen con los comandos nslookup y con ping. El correo puede salir vía otro host, así que compruebe los records MX en el espacio de nombre de dominio. Compruebe que exista el usuario.

Intente efectuar un finger al host de destino. Intente un telnet al puerto 25 y use el comando VRFY, entonces use el comando EXPN para las listas de correo.

Intente enviar correo de prueba manualmente. Primero, telnet al puerto 25. Luego escriba los siguientes comandos “HELO”, “MAIL”, “RCPT”, y “DATA”. Si todo esto falla, envíe un correo al postmaster.

## RESOLUCION DE PROBLEMAS TCP/IP

Las NICs (Tarjetas de Interfaz de Red) le ofrecen a los sistemas hablar uno con otro en una red. El nivel del hardware es el primero de la resolución de problemas en un NIC. Consulte el manual del usuario que viene con la tarjeta de red o el módem para asegurarse que esten correctamente instalado y configurados. Después de probar el hardware, un número de programas están disponibles para localizar averías enle interfaz incluyendo Ismod, insmod, ifconfig, ping, y tcpdump.

Los asuntos siguientes se discuten en esta sección:

- Módulos de Kernel
- Comandos de Resolución de Problemas de la Red
- Archivos Útiles de la Red
- ICMP (Internet Control Message Protocol)
- Otras Fuentes de Información

***Módulos de y el Kernel***

Antes de que usted pueda utilizar su interfaz de la red, el módulo correcto debe ser cargado en el núcleo para que reconozca la tarjeta. En GNU/Linux, los drivers están en forma de módulos del kernel. Para una vez que se cargue un módulo, hay un número de comandos para visualizar y manipular los módulos.

Hay muchos utilitarios para visualizar la información de configuración de la red y monitorear la infformación del hardware, pero el kernel proporciona una manera fácil de tener acceso a esta información atraves del sistema de archivos virtual proc en /proc.

En esta sección se discuten ambos los módulos del kernel y el sistema de archivos virtual proc.

***Módulos del Kernel***

Los comandos modprobe, lsmod, insmod y rmod son útiles para manipular módulos para instalar y configurar las interfaces de red. Todos estos comandos requieren la autenticación del super usuario. Escriba el siguiente comando para visualizar la información acerca de los módulos cargados:

***#/sbin/lsmod***

Este comando despliega la información del nombre, el tamaño, la cuenta del uso, y la lista de los módulos referidos.

| ***Module*** | ***Size*** | ***Used by*** | ***Not tainted*** |
|---|---|---|---|
| ***appletalk*** | ***23884*** | ***12*** | ***(autoclean)*** |

```
8139too         17024   1
mii             3976    0       (8139too)
e100            51428   1
ipt_LOG         4416    1       (autoclean)
```

El comando modprobe puede cargar un conjunto de módulos, un solo módulo, una pila de módulos dependientes o todos los módulos marcados con una etiqueta particular. El modprobe primero busca en los directorios que contienen módulos compilados para la versión más reciente del kernel de GNU/Linux.

El comando modprobe puede explorar para los módulos disponibles y carga automáticamente cualquier módulo bajo necesitado en un apilado del módulo. La línea de comando siguiente inicia el modprobe para enumerar todos los módulos disponibles:

```
# /sbin/modprobe  -l
```

El insmod instala un módulo cargable en el kernel en ejecusión. Este hace el intento de vincular (linkear) un módulo al núcleo corriente mirando la tabla de símbolo exportada del núcleo. El siguiente comando intenta instalar el módulo para la tarjeta de Ethernet Realtek 8139too:

```
# /sbin/modprobe 8139too
```

El comando rmmod descarga los módulos cargado en el kernel. Este comando quita todos los módulos cargado en el kernel.

```
# /sbin/modprobe -a
```

Después de instalar correctamente el módulo apropiado para el interfaz de la red, usted puede utilizar el comando ifconfig de configurar y de modificar los interfaces en tiempo real.

***El Sistema de Archivos Virtual proc***

El directorio /proc en un sistema GNU/Linux contiene los archivos virtuales creados por el kerne! de modo que usted y el sistema puedan supervisar qué está sucediendo actualmente en su sistema. Algunos de estos archivos son útiles para analizar y diagnosticar averías de interfaces de red.

Escriba la siguiente línea:

```
# cat /proc/net/dev
```

Esto da salida a las líneas con la tarjeta Ethernet listada. Si la salida no es lo que se esperaba, entonces su kernel de GNU/Linux no sabe de la existencia de su adaptador de red. Una razón posible para que el kernel no reconozca la tarjeta Ethernet es porque tenga un conflicto con la interrupción o de dirección. Para ver una lista de las interrupciones y de las direcciones de I/O (Entrada/Salida), escriba los siguientes comandos:

```
# cat  /proc/interrupts
# cat  /proc/ioports
```

El comando cat /proc/interrupts producirá una salida similar a la siguiente:

```
[root@root]# cat /proc/interrupts
        CPU0
 0:   10269073        XT-PIC  timer
 1:          8        XT-PIC  keyboard
 2:          0        XT-PIC  cascade
 8:          1        XT-PIC  rtc
11:    3876069        XT-PI
```

La interrupción es una señal enviada desde un dispositivo a la entrada del procesador de que un servicio es necesario. A menudo las interrupciones son llamados IRQs. La lista muestra que el adaptador de red eth0 situado en el interruptor 11 y que ningún otro dispositivo está compartiendo este número de la interrupción.

```
[root@proxy-ap root]# cat /proc/ioports
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0070-007f : rtc
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02f8-02ff : serial(auto)
0376-0376 : ide1
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial(auto)
0cf8-0cff : PCI conf1
1000-101f : Intel Corp. 82557/8/9 [Ethernet Pro 100]
  1000-101f : e100
1020-103f : Intel Corp. 82371AB/EB/MB PIIX4 USB
  1020-103f : usb-uhci
1040-104f : Intel Corp. 82371AB/EB/MB PIIX4 IDE
  1040-1047 : ide0
  1048-104f : ide1
1400-14ff : Accton Technology Corporation SMC2
```

La dirección 1000-101f de I/O esta asignada solamente al eth0, así demostrando que no hay ningún conflicto.

## Comandos de Resolución de Problemas de Redes

Utilice los siguientes comandos para la resolución de problemas en la red:

- ifconfig
- ping
- tcpdump
- traceroute
- netstat
- netperf
- arp
- nslookup
- hostname

***ifconfig***

El comando Ifconfig es un programa que gestiona los dispositivos de red conectados a tu equipo y los enlaza con interfaces de red. Ejemplos de dispositivos de red posibles son tarjetas de red (interfaz ethX), modems (pppX), puerto paralelo (plipX), infrarrojos ... Una vez que tengamos la tarjeta de red bien instalada, (es decir, cargados los módulos correspondientes o compilado en el kernel el soporte para la tarjeta) tendremos que crear un interfaz que será por el que se envíe la información a los equipos conectados a

este dispositivo de red. Para esto ejecutamos el comando ifconfig.

El Ifconfig es una utilidad de línea de comandos que permite obtener y configurar las interfaces de red de un equipo. Si no se proporcionan argumentos, ifconfig muestra el estado de las interfaces de red que se encuentran activas. Si se proporciona una interfaz como argumento, ifconfig muestra el estado de dicha interfaz. Si se utiliza con la opción -a, muestra el estado de todas las interfaces, incluso aquellas que se encuentren desactivadas. Para configurar una interfaz se debe utilizar el formato:

***ifconfig <interfaz> <familia> <dir_ip> netmask <máscara> broadcast <dir_broadcast> up***

- **En interfaz**

  Debe proporcionarse el nombre de la interfaz de red que se desea configurar. Generalmente el nombre de interfaz se forma a partir de un nombre de manejador seguido de un número de unidad. El nombre de manejador para redes Ethernet es eth y las unidades comienzan a numerarse a partir de 0 hasta el número de interfaces existentes del mismo tipo menos uno (eth0, eth1, etc.).

- **En familia**

  Se debe proporcionar el nombre de una familia de direcciones soportada por el sistema. Este nombre se utilizará para decodificar y mostrar en un formato inteligible todas las direcciones de protocolo. Las familias de direcciones más comúnmente utilizadas son inet para TCP/IP, inet6 para IP versión 6 e ipx para Novell IPX.

- **dir_ip**

  Es la dirección IP con que se desea configurar la interfaz de red.

- **máscara**

  Establece la máscara de red que se desea utilizar para la interfaz. Si no se proporciona este valor, se utilizarán las máscaras de red por defecto para direcciones clase A, B o C en función de la dirección IP con que se esté configurando esta interfaz.

- **dir_broadcast**

  Al proporcionar una dirección de broadcast con la opción broadcast, se indica a la interfaz de red que se desea que habilite el modo de broadcast dirigido y que contemple dicha dirección. La dirección de broadcast dirigido a una red se determina a partir de la dirección IP de cualquiera de los equipos pertenecientes a dicha red y su máscara de red, ya que se forma a partir la dirección de red poniendo '1's en la parte de dirección correspondiente a equipos.

El comando ifconfig presenta muchas otras opciones que pueden ser consultadas mediante la ayuda en línea de GNU/Linux (***man ifconfig***). Una de ellas, útil para el desarrollo de la práctica, es la opción mtu valor, que permite establecer la unidad máxima de transferencia (MTU, del inglés Maximum Transfer Unit) que se desea utilice la interfaz que está configurándose. Antes de reconfigurar una interfaz de red, es conveniente desactivarla utilizando la opción down. Utilice para ello el formato ***ifconfig interfaz down***

Para ver un listado de los diversos interfaces se demuestra aquí debajo:

```
k1k1@kikla:~$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0B:CD:36:04:2A
          inet6 addr: fe80::20b:cdff:fe36:42a/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:23670 (23.1 KiB)
          Interrupt:10 Base address:0x4000

eth1      Link encap:UNSPEC  HWaddr 00-0B-CD-71-A0-35-D7-22-00-00-00-00-00-00-00-00
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:71 dropped:71 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

```
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth2    Link encap:Ethernet  HWaddr 00:09:5B:54:89:26
        inet addr:10.0.0.106  Bcast:10.0.0.255  Mask:255.255.255.0
        inet6 addr: fe80::209:5bff:fe54:8926/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:7153 errors:0 dropped:0 overruns:0 frame:0
        TX packets:5212 errors:5 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:3487577 (3.3 MiB)  TX bytes:1293400 (1.2 MiB)
        Interrupt:3 Base address:0x100

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:167 errors:0 dropped:0 overruns:0 frame:0
        TX packets:167 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:28096 (27.4 KiB)  TX bytes:28096 (27.4 KiB)

sit0    Link encap:IPv6-in-IPv4
        inet6 addr: ::127.0.0.1/96 Scope:Unknown
        UP RUNNING NOARP  MTU:1480  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:69 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Si el PPP esta configurado en el sistema, será incluido en la lista de interfaces. Los siguientes son algunos comandos que usted puede utilizar para ver si la dirección IP y mascaras de red son las correcta asociadas con la tarjeta de Ethernet.

El formato del comando es:
***/sbin/ifconfig interface opciones***

Una de las aplicaciones del comando ifconfig es el de iniciar y detener una interface Ethernet usando la opciones up y down.

Por ejemplo:
```
# /sbin/ifconfig eth0 up
# /sbin/ifconfig eth0 down
```

Si falta el interfaz del loopback, escriba:
```
# /sbin/ifconfig lo 127.0.0.1
```

Si le falta el adaptador de la red (tarjeta de Ethernet), escriba:
```
# ifconfig eth0 192.168.2.7 up
```
La dirección IP interna ahora es asociada con esta tarjeta de Ethernet.

Para incluir un mascara de red, escriba:
```
# ifconfig eth0  192.168.2.7  netmask 255.255.255.0
```

Si la dirección de red es una dirección de la clase C, el ifconfig asignará automáticamente la netmask 255,255,255.0.

Si estos comandos todavía no consiguen ingresar la computadora en la red, entonces investigue dentro del kernel de su GNU/Linux para ver qué dispositivos tiene.

***El comando ping***

El utilitario ping usa datagramas de ECHO_REQUEST para obtener un ICMP ECHO_RESPONSE de un host o de una pasarela (gateway). Los ping incluyen el número de bytes devueltos desde la interfaz, la secuencia y el tiempo de ida y vuelta. Esta utilidad proporciona el mejor método para confirmar si un adaptador de red está configurado correctamente y si el enrutamiento esta establecido correctamente. La primera cosa quedebemos hacer es ping al interfaz de loopback.

Escriba el siguiente comando para efectuar un ping al interfaz loopback:
***# ping 127.0.0.1***

Esto debe producir una respuesta similar a la siguiente:
***k1k1@kikla:~$ ping 127.0.0.1***
***PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.***
***64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.067 ms***
***64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.079 ms***
***--- 127.0.0.1 ping statistics ---***
***3 packets transmitted, 3 received, 0% packet loss, time 2000ms***
***rtt min/avg/max/mdev = 0.067/0.073/0.079/0.005 ms***

El ping puede también seguir tráfico de la red en otras máquinas conectadas a la red. El comando ping imprime una línea por cada respuesta recibida. El comando ping continuará recibiendo paquetes hasta que el usuario termine (generalmente por un CONTROL+C). El comando ping es útil para observar el tiempo de respuesta de los paquetes o ver si se saltan las secuencias de los números en serie. Estos datos son importantes en la investigación de pérdida de paquete y tráfico de la red.

En la siguiente tabla se enumeran varias opciones del comando ping.

| Opción | Descripción |
|---|---|
| -c #número | Detiene el ping después de enviar y recibir #número de paquetes ping. |
| -f flood ping | Da salida de paquetes ping lo más rápido posible o 100 por segundo. Sólo el root puede usar esta opción. |
| -i wait | Esta opción espera (waits) los segundos indicados por el valor wait entre los paquetes enviados. |
| -s patketsize | Esta específica el número de bytes a ser enviado. Por defecto es de 56, pero con el cabezal ICMP la data se con vierte en 64 bytes. |

***El Comando tcpdump***

El Tcpdump es un programa cuya utilidad principal es analizar el tráfico que circula por la red. Se apoya en la librería de captura pcap, la cual presenta una interfaz uniforme y que esconde las peculiaridades de cada sistema operativo a la hora de capturar tramas de red. Para seguir el manual es necesario unos conocimientos basicos del protocolo TCP/IP, remitiéndome al TCP/IP Illustrated, Volumen 1 de Stevens, para quien esté interesado.

Aunque viene incluido con la mayoría de las distribuciones de Linux, sus fuentes pueden encontrarse en www.tcpdump.org. Lo ejecutamos simplemente escribiendo:

***# tcpdump***

Pero esto no es lo normal, empecemos a introducir opciones y comentar para que es cada una. Algunos ejemplos rapidos de tcpdump:

***# tcpdump –i eth0 –c 100 –s 500***

Nos sacará por pantalla los primeros 100 paquetes (-c 100) que pasen por la interfaz eth0 (-i eth0) con un tamaño maximo de paquete de 500 bytes (-s 500)

***# tcpdump –qec 1***

Tenemos varias formas de sacar paquetes por pantalla: -q (quiet o silencioso (poca información)) y –v y –vv que van de menos a más información. La opción –e nos sacará las direcciones mac origen/destino y –c 1 nos sacará solo el primer paquete como ya hemos explicado antes.

Que tipo de tráfico podemos capturar: Podemos capturar tráfico basado en:

- Direcciones
- Protocolos
- Puertos
- Características de paquetes
- Combinacion de todos estos

***Filtrando por Direcciones***

***# tcpdump host ivelis***

Nos sacará toda la información relativa al host '***ivelis***'.

***# tcpdump dst host ivelis***

Nos sacará toda la información donde el host destino sea 'ivelis'

***# tcpdump ether src host 0:a0:3b:3:e1:1d***

Nos sacará toda la información donde la tarjeta de red de origen sea la dirección mac: 0:a0:3b:3:e1:1d (Sintaxis: ether host )

***# tcpdump -c 100 net 192.168.1.0 mask 255.255.255.0***

Mostrará los primeros 100 paquetes (-c 100) de la red 192.168.1.0 (net 192.168.1.0) con una mascara de red de 255.255.255.0 (mask 255.255.255.0)

***Filtrando por Protocolos***

***# tcpdump upd***

Mostrará todos los paquetes que viajen por udp

***# tcpdump –i eth0 –c 1000 –s 300 udp***

Mostrará los primeros 1000 (-c 1000) paquetes de la interfaz eth0 (-i eth0) con un tamaño de paquete maximo de 300 bytes (-s 300) en el protocolo udp.

***# tcpdump port 23***

Mostrará todos los paquetes que vayan por el puerto 23

***Mezclando filtros***

***# tcpdump not port 22***

Mostrará toda la información excepto la que vaya por el puerto 22

***# tcpdump port 23 and host ivelis***

Mostrará toda la información que vaya por el puerto 23 y por el host ivelis

***# tcpdump not "(port 23 and host ivelis and host miguel)"***

Mostrará toda la información que ni vaya por el puerto 23, ni por el host ivelis ni por el host miguel

***# tcpdump dst host ivelis and dst port 80***

Mostrará toda la información que vaya al host ivelis y al puerto destino 80 (dst port 80)

***# tcpdump –i etho –c 100 –s 1000 src host miguel and "(dst host ivelis or dst host crithian)" and dst port 22***

Como puede apreciar, mezclando información podemos tener filtros largos y todo lo complicados que queramos… este filtro nos mostrará los primeros 100 paquetes que pasen por la interfaz eth0 (-i eth0) con tamaño maximo de paquete de 1000 bytes (-s 1000) que vengan del host miguel (src host miguel) y que vayan o al host ivelis o al host cristhian "(dst host ivelis or dst host cristhian)" al puerto 22 (dst port 22).

***El Comando traceroute***

Esta orden se utiliza para imprimir la ruta que los paquetes siguen desde nuestro sistema hasta otra máquina; para ello utiliza el campo TTL (Time To Live) del protocolo IP, inicializándolo con valores bajos y aumentándolo conforme va recibiendo tramas ICMP de tipo TIME/SMALL>_EXCEEDED. La idea es sencilla: cada vez que un paquete pasa por un router o una pasarela, esta se encarga de decrementar el campo TTL en una unidad; en el caso de que se alcance un valor 0, se devuelve un paquete TIME/SMALL>_EXCEEDED y se descarta la trama. Así, traceroute inicializa a 1 este campo, lo que ocasiona que el primer router encontrado ya devuelva el mensaje de error; al recibirlo, lo inicializa a 2, y ahora es el segundo router el que descarta el paquete y envía otro mensaje de error, y así sucesivamente. De esta forma se va construyendo la ruta hasta un determinado host remoto:

```
# traceroute www.codigolibre.org
 1  * * *
 2  209.58.3.52 (209.58.3.52)  20.45 ms  25.05 ms  31.593 ms
 3  192.168.102.1 (192.168.102.1)  17.098 ms  15.996 ms  15.178 ms
 4  172.22.0.205 (172.22.0.205)  15.444 ms  17.175 ms  15.143 ms
 5  172.22.0.253 (172.22.0.253)  24.353 ms  25.77 ms  24.13 ms
 6  172.22.0.133 (172.22.0.133)  21.846 ms  26.21 ms  21.839 ms
 7  hme0-bavaro.codetel.net.do (196.3.74.5)  49.477 ms  49.497 ms  48.532 ms
 8  ge-1-0-0.r01.miamfl02.us.bb.verio.net (129.250.11.101)  52.83 ms  50.249 ms  50.725 ms
 9  p4-0-0-0.r00.miamfl02.us.bb.verio.net (129.250.3.184)  48.745 ms  50.081 ms  48.285 ms
10  p4-4-2-0.r01.dllstx09.us.bb.verio.net (129.250.5.58)  92.83 ms  96.722 ms  94.801 ms
11  so-2-3-0.edge2.dallas1.level3.net (4.68.127.41)  93.029 ms  100.019 ms  92.208 ms
12  so-7-0-0.bbr2.dallas1.level3.net (4.68.96.121)  110.301 ms  117.371 ms  112.163 ms
13  4.68.122.11 (4.68.122.11)  88.584 ms 4.68.122.43 (4.68.122.43)  86.717 ms  87.021 ms
14  4.78.224.74 (4.78.224.74)  87.983 ms  90.102 ms  86.997 ms
15  206.123.64.42 (206.123.64.42)  90.201 ms  88.392 ms  90.278 ms
16  abiertos.org (206.123.67.148)  86.21 ms  87.712 ms  86.113 ms
```

El comando traceroute se utiliza para realizar pruebas, medidas y administración de una red; introduce mucha sobrecarga, lo que evidentemente puede acarrear problemas de rendimiento, llegando incluso a negaciones de servicio por el elevado tiempo de respuesta que el resto de aplicaciones de red pueden presentar. Además, se trata de un programa contenido en un fichero setuidado, por lo que es interesante resetear el bit de setuid de forma que sólo el root pueda ejecutar la orden: hemos de pensar que un usuario normal rara vez tiene que realizar pruebas sobre la red, por lo que el bit setuid de traceroute no es más que un posible problema para nuestra seguridad; aunque con ping sucede lo mismo (es un fichero setuidado), que un usuario nece-

site ejecutar traceroute es menos habitual que que necesite ejecutar ping (de cualquier forma, también podríamos resetear el bit setuid de ping).

El formato de comando para Linux es:

***$ traceroute dirección_ip***

En el ejemplo, dirección_ip identifica el sistema remoto. En la sguiente tabla enumeramos varias opciones de la línea de comandos.

| Opción | Descripción |
|---|---|
| -f | Establece el valor TTL inicial en el primer paquete saliente |
| -F | Estable el bit del valor "don't fragment" |
| -g host-list | Específica la ruta a un gateway "loose-source" |
| -l | Opción usa ICMP ECHO en ves de datagramas UDP |
| -m | Específica el máximo número de saltos que el utilitario buscará; por defecto es 30 |
| -n | Nombre del host no se resolverá a direcciones en cada salto . |
| -p | Establece el número de puerto base UDP usado en las pruebas (por defecto es 33434). |
| -q | Número de consultas para el número de peticiones del número de paquetes que deben ser enviados por cada valor TTL para recibir el valor RTT por cada salto |
| -r | Sobre pasa las tablas de enrutamiento normal y envía directamente a un host conecta do a la red. |
| -s | Utiliza la dirección IP proveída en los paquetes de prueba que envía |
| -t | Específica el valor ToS en los paquetes de prueba a enviar |
| -w timeout | Específica el tiempo en segundos a esperar por cada respuesta. Por defecto es 5 segundos. |

Ejemplos de traceroute desde la línea de comando:

***# traceroute nombre-host***

dará lugar a la siguiente respuesta de un solo salto debido a que está en la misma subnet:

***$ traceroute ivelis***
***traceroute to ivelis (10.0.0.11), 30 hops max, 40 byte packets***
***1 ivelise (10.0.0.11) 22.323 ms 2.63 ms 0.707 ms***

Si no hay respuesta de un enrutador dentro de un intervalo timeout de 5 segundos, se imprime una "*" para ese paquete de prueba. Este intervalo de timeout se puede cambiar con la opción -w según se explica en la tabla de opciones.

El comando traceroute es para el uso de encontrar fallas en la red, y debe ser usado sólo por la gerencia. Debe ser utilizado sobre todo para el aislamiento manual de fallas ya que este solamnete procura seguir la trayectoria que viaja el paquete. Este impone una carga adicional a la red con los paquetes a la destinación con diversos valores TTL y por lo tanto, no es sabio utilizar el traceroute durante tiempo normales de operaciones o escribir scripts de tareas automatizadas ya que inundan la red. Impone una carga adicional ante la red con los paquetes a la destinación con diversos valores de la TTL, y por lo tanto, no es sabio utilizar el traceroute durante operaciones normales o de las escrituras automatizadas pues inunda la red.

***El Comando netstat***

Esta orden se utiliza para visualizar el estado de diversas estructuras de datos del sistema de red, desde las tablas de rutado hasta el estado de todas las conexiones a y desde nuestra máquina, pasando por las tablas ARP, en función de los parámetros que reciba.

En temas referentes a la seguridad, netstat se suele utilizar, aparte de para mostrar las tablas de rutado de ciertos sistemas (con la opción -r, como hemos visto antes), para mostrar los puertos abiertos que escuchan peticiones de red y para visualizar conexiones a nuestro equipo (o desde él) que puedan salirse de lo habitual. Veamos un ejemplo de información mostrada por netstat:

```
# netstat -P tcp -f inet -a
TCP
   Local Address       Remote Address      Swind Send-Q Rwind Recv-Q State
-------------------- -------------------- ----- ------ ----- ------ -----------
      *.*                  *.*                0      0      0      0 IDLE
      *.sunrpc             *.*                0      0      0      0 LISTEN
      *.*                  *.*                0      0      0      0 IDLE
      *.32771              *.*                0      0      0      0 LISTEN
      *.ftp                *.*                0      0      0      0 LISTEN
      *.telnet             *.*                0      0      0      0 LISTEN
      *.finger             *.*                0      0      0      0 LISTEN
      *.dtspc              *.*                0      0      0      0 LISTEN
      *.lockd              *.*                0      0      0      0 LISTEN
      *.smtp               *.*                0      0      0      0 LISTEN
      *.8888               *.*                0      0      0      0 LISTEN
      *.32772              *.*                0      0      0      0 LISTEN
      *.32773              *.*                0      0      0      0 LISTEN
      *.printer            *.*                0      0      0      0 LISTEN
      *.listen             *.*                0      0      0      0 LISTEN
      *.32774              *.*                0      0      0      0 LISTEN
      *.*                  *.*                0      0      0      0 IDLE
      *.6000               *.*                0      0      0      0 LISTEN
      *.32775              *.*                0      0      0      0 LISTEN
localhost.32777      localhost.32775        32768      0  32768      0 ESTABLISHED
localhost.32775      localhost.32777        32768      0  32768      0 ESTABLISHED
localhost.32780      localhost.32779        32768      0  32768      0 ESTABLISHED
localhost.32779      localhost.32780        32768      0  32768      0 ESTABLISHED
localhost.32783      localhost.32775        32768      0  32768      0 ESTABLISHED
localhost.32775      localhost.32783        32768      0  32768      0 ESTABLISHED
localhost.32786      localhost.32785        32768      0  32768      0 ESTABLISHED
localhost.32785      localhost.32786        32768      0  32768      0 ESTABLISHED
localhost.32789      localhost.32775        32768      0  32768      0 ESTABLISHED
localhost.32775      localhost.32789        32768      0  32768      0 ESTABLISHED
localhost.32792      localhost.32791        32768      0  32768      0 ESTABLISHED
localhost.32791      localhost.32792        32768      0  32768      0 ESTABLISHED
localhost.32810      localhost.6000         32768      0  32768      0 ESTABLISHED
localhost.6000       localhost.32810        32768      0  32768      0 ESTABLISHED
anita.telnet         luisa.2039             16060      0  10136      0 ESTABLISHED
anita.telnet         ltorvalds.linux.org.1068 15928    0  10136      0 ESTABLISHED
localhost.32879      localhost.32775        32768      0  32768      0 TIME_WAIT
      *.*                  *.*                0      0      0      0 IDLE
```

Por un lado, en este caso vemos que hay bastantes puertos abiertos, esto es, escuchando peticiones: todos los que presentan un estado LISTEN, como telnet, finger o smtp (si es un servicio con nombre en /etc/services se imprimirá este nombre, y si no simplemente el número de puerto). Cualquiera puede conectar a este servicio (como veremos en el siguiente punto) y, si no lo evitamos mediante TCP Wrappers, utilizarlo para enviarle peticiones.

Aparte de estos puertos a la espera de conexiones, vemos otro gran número de conexiones establecida entre nuestro sistema y otros (como antes hemos dicho, desde nuestro equipo o hacia él); casi todas las esta-

blecidas (estado ESTABLISHED) son de nuestra máquina contra ella misma, lo que a priori no implica consecuencias de seguridad. Otra de ellas es desde un equipo de la red local contra nuestro sistema, lo que también es bastante normal y no debe hacernos sospechar nada; sin embargo, hay una conexión que sí puede indicar que alguien ha accedido a nuestro sistema de forma no autorizada: si nos fijamos, alguien conecta por telnet desde la máquina ltorvalds.linux.org. Es raro que tengamos a un usuario allí, por lo que deberíamos monitorizar esta conexión y las actividades que esta persona realice; es muy probable que se trate de alguien que ha aprovechado la inseguridad de ciertos sistemas para utilizarlos como plataforma de ataque contra nuestros UNiX.

La tabla siguiente enumera las opciones comúnmente usadas con el comando del netstat.

| Opción | Descripción |
|---|---|
| -a | Muestra el estado de todos los sockets |
| -c | Refresca toda la información cada segundo |
| -n | Reporta las direcciones de Internet y los números de puertos como números y no como nombre de hosts ni como servicios |
| -r | Desplega las tablas de enrutamiento del sistema |
| -s | Prové estadisticas de los paquetes procesados por su sistema |
| -i | Muestra el estado de las interfaces que han sido autoconfiguradas o una interface particular si es especificada |
| -1 | Desplega los puertos actualmente escuchando y los sockets |

El comando del netstat de Linux proporcionará mucha estadística sin ningunas opciones. Por ejemplo, desde la línea de comandos de GNU/Linux escriba:

***# netstat***

Esto le dará una respuesta (sólo se presenta una parta para breviedad) como la siguiente:

```
k1k1@kikla:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q  Send-Q  Local Address           Foreign Address         State
tcp   0       0       10.0.0.182:43302       abiertos.org:www        ESTABLISHED
tcp   0       0       10.0.0.182:43298       abiertos.org:www        ESTABLISHED
udp   0       0       localhost.localdom:1024 localhost.localdom:1024 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State      I-Node   Path
unix  10     [ ]     DGRAM                 5645     /dev/log
unix  2      [ ]     DGRAM                 5852     @/var/run/hal/hotplug_socket
unix  2      [ ]     DGRAM                 1746     @udevd
unix  2      [ ]     DGRAM                 12248
unix  3      [ ]     STREAM     CONNECTED  10382    /tmp/.ICE-unix/4605
unix  3      [ ]     STREAM     CONNECTED  10381
unix  3      [ ]     STREAM     CONNECTED  10375    /tmp/.X11-unix/X0
```

***El Comando netperf***

Netperf es un benchmark (prueba de rendimiento) que puede ser usada para localizar averías y problemas en una red. La información proporcionada por netperf puede ser utilizada para localizar congestión en una red y entonces se pueden tomar las medidas apropiadas para aliviar el problema.

El programa netperf proporciona información en una gran variedad de asuntos, incluyendo ambas prueba de flujo y de petición/respuesta y puede usar varios diversos protocolos de red, incluyendo el TCP y UDP. Un factor importante que se debe observar es que para probar una interfaz de red con excepción de la interfaz loopback, lo, el netperf debe estar instalado en otra máquina en la red.

Hay dos programas separados que conforman a netperf, el programa cliente, nerperf, que es el responsa-

ble de iniciar una conexión a la otra parte que es el servidor de netperf, llamado netserver. Puesto que netperf contiene el netserver, hay dos maneras posibles de instalarlo. Para los usuarios con el acceso a la cuenta de root, el netserver se puede instalar como demonio invocado por el demonio de Internet, el inetd. Si no, el netserver se puede instalar como demonio independiente.

***Usar netperf***

Como se mencionó previamente, netperf puede realizar diversos tipos de pruebas. Los tipos más comunes de pruebas lo más probable serán para TCP y UDP. Hay varias opciones de la línea de comandos para cambiar los tipos y las especificaciones de las pruebas que se desean realizar. Algunas de las opciones de la línea de comandos son:

| | |
|---|---|
| -s size | Específica el tamaño del socket del buffer local de enviar y recibir |
| -S size | Efectuá la operación de -s para la máquina remota |
| -m size | Envia los paquetes de tamaño size en bytes |
| -M size | Recibe paquetes de tamaño size en bytes; como -m para sistema remoto |
| -t test | Específica el tipo de prueba a efectuar |
| -H host | Se conecta a host para poner a prueba la red |
| -l length | Efectuá la prueba por length secongundos |
| -h | Desplega un mensaje de uso |

El comando /opt/netperf/netperf produce una prueba del flujo TCP. Esta prueba por general se realiza en el interfaz del loopback y tiene una duración de 10 segundos. La prueba envía un mensaje de un puerto a otro. El tamaño del mensaje se mide en el puerto de inicio en el puerto final. El rendimiento de procesamiento se mide en kilobytes por segundo. Aquí le presentamos un método rápido para comprobar si el programa del netperf está trabajando correctamente:

***k1k1@kikla:~$ /opt/netperf/netperf -H host -t type***

Aquí el host es el destino y los tipos de pruebas son especificados por host y type (tipo), respectivamente. Para poder conectarse con el otro host, es importante recordar que el host de destino debe estar ejecutando netserver. Algunas de las pruebas que se pueden especificar con la opción -t son:

| | |
|---|---|
| UDP_STREAM | Efectuá una prueba de Flujo UDP |
| TCP_STREAM | Efectuá una prueba de Flujo TCP |
| UDP_RR | Efectuá una prueba de Petición/Respuesta UDP |
| TCP_RR | Efectuá una prueba de Petición/Respuesta TCP |

Hay varias otras pruebas que pueden ser realizadas por netperf; las que están enumerados anteriormente son generalmente las más comunes.

***El Comando arp***

El comando arp exhibe y modifica las tablas de conversión de dirección de Internet-a-Ethernet usadas por el ARP (Address Resolution Protocol). El formato de comando es: ***# arp options***

Cuando esté es utilizado sin ningunas opciones, el comando arp exhibe la información actualmente almacenada. El siguiente es un ejemplo de como usar el comando arp con su resultado:

***kikla:/home/k1k1# arp***

| *Address* | *HWtype* | *HWaddress* | *Flags Mask* | *Iface* |
|---|---|---|---|---|
| *10.0.0.1* | *ether* | *00:08:C7:F3:E6:E4* | *C* | *eth2* |

La opción -a de arp despliega la información en el formato raw:

```
kikla:/home/k1k1# arp -a
? (10.0.0.1) at 00:08:C7:F3:E6:E4 [ether] on eth2
```

Puede suprimir entradas del cache de arp usando la opción -d y especificando un nombre de host:

```
kikla:/home/k1k1# arp -d  nombre_host
```

***El Comando nslookup***

El comando nslookup es un programa interactivo para efectuar preguntas a los servidores de nombre de dominio del Internet. El usuario tiene la opción de solicitar a un servidor de nombres en específico para que le proporcione la información sobre un host en particular o para conseguir una lista de todos los host en un dominio.

El formato del comando nslookup es:

```
# nslookup  opciones    dirección
```

El siguiente es un ejemplo de como usar el comando nslookup:

```
kikla:/home/k1k1# nslookup miguel
Server:          usuarios.codigolibre.org
Address:         192.168.2.53
Name:            miguel.codigolibre.org
Address:         192.168.2.129
```

Para obtener una lista de todos los nodos en un dominio en particular, ejecute la siguiente secuencia de comandos:

```
# nslookup
> ls codigolibre.org
[ns.codigolibre.org]
$ORIGIN  codigolibre.org
@          ID      IN      A       192.168.2.254
usuarios   ID      IN      A       192.168.2.53
miguel     ID      IN      A       192.168.2.125
ns         ID      IN      A       192.168.2.254
>exit
```

***El Comando hostname***

El comando hostname imprime el nombre del host actual, parecido al presentado por el prompt del sistema, si así esta configurado, al inició de sesión o login. Aquí se le presenta un ejemplo de usar el comando hostname:

```
kikla:/home/k1k1# hostname
kikla
```

**Ejercicio 5-1: Instale y Configure una tarjeta de Ethernet Como Interfaz de Red**

En este ejercicio vamos a instalar correctamente una tarjeta Ethernet como interfaz de red. En este ejercicio, la tarjeta Ethernet es una 3COM 3C905. No se proporcionan soluciones para este ejercicio.

1. Ingrese al sistema como root.

2. Para buscar el módulo para utilizar para el dispositivo de red 3COM 3C905, escriba el siguiente comando para buscar el módulo apropiado:

   ```
   /sbin/modprobe -l
   ```

   Para enumerar los módulos una página a la vez, escriba:

***/sbin/modprobe -l | more***

3. El módulo en este caso será 3c59x.o. Escriba lo siguiente para instalar el módulo:
   ***/sbin/insmod 3c59x***

4. Para ver si el módulo ha sido instalado correctamente, digite:
   ***/sbin/lsmod***

5. Después de asegurarse que el módulo está presente, utilize el comando ifconfig para configurar la tarjeta de Ethernet. Asegúrese de que la tarjeta Ethernet está alias al interfaz correcto (eth0, eth1, eth2, etc.) en /etc/conf.modules en el archivo, sólo debe escribir:
   ***# cat /etc/conf.modules***

   Si el interfaz no es el deseado, utilice un editor de texto, como pico y modifiqúelo. Ahora utilice el ifconfig para asignarle una Dirección IP:
   ***/sbin/ifconfig eth0 192.168.2.15 up***
   Esta sentencia asigna la IP ADDRESS 192.168.2.7 a la interfaz eth0 y la activa (up).

6. Ahora es necesario manipular la tabla de enrutamiento para establecer las rutas estáticas en la red:
   ***/sbin/route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0***

   Esta sentencia establece una ruta a la red 192.168.2.x atraves de la interfaz eth0

7. Por último utilice el comando ifconfig para ver si la tarjeta Ethernet ahora se está presentando activa en la red:
   ***/sbin/ifconfig***

## Archivos de Red Utiles

Todo administrador de una red TCP/IP debe conocer bien los siguientes archivos de red. Entre los archivos claves de la red se incluyen:

- protocols
- services
- inetd.conf

***El Archivo protocols***

Cada línea del archivo protocolos describe los protocolos usados en la capa de Internet de la pila del protocolo TCP/IP según lo definido por RFC 1700 (Assigned Numbers). Cada línea se ajusta al siguiente formato:

***nombre-oficial-del-protocolo protocol # aliases***

El archivo de los protocolos es almacenado en el directorio /etc y se puede corregir fácilmente usando un editor de texto. Si usted desarrolla su propio protocolo que utiliza la interface raw de sockets, este deberá ser listado en el archivo protocol(s).

El número listado para cada protocolo se incluye en el campo Protocolo del cabezal IP. Ese número define el próximo nivel de protocolo para recibir el campo Data en el destino. El archivo exhibe la siguiente información (seleccionada), que variará dependiendo de la personalización para requisitos particulares:

***$ cat /etc/protocols***
***#***
***# Internet protocols***

```
#
# $GNU/Linux:src/etc/protocols,v 1.14 2000/09/24 11:20:27 asmodai Exp $
#       from: @(#)protocols     5.1 (Berkeley) 4/17/89
#
# See also http://www.isi.edu/in-notes/iana/assignments/protocol-numbers
#
ip      0       IP              # Internet protocol, pseudo protocol number
#hopopt 0       HOPOPT          # hop-by-hop options for ipv6
icmp    1       ICMP            # Internet control message protocol
igmp    2       IGMP            # Internet group management protocol
ggp     3       GGP             # gateway-gateway protocol
```

***El Archivo services***

El archivo services contiene los números de los puertos de los servicios well-known (servicios bien conocidos) según están definidos por el RFC 1700 (Assigned Numbers). Cualquier aplicación que desea reservar un número de puerto debe especificarlo en el archivo services.

NOTA: Con simplemente listar una aplicación en el archivo de los servicios no significa que la aplicación estará soportada. El nombre del servicio y el número del puerto son una parte pequeña solamente de una aplicación funcional.

El formato de cada línea es como sigue:
***nombre-del-servicio puerto/protocolo aliases # descripción/comentarios***

El archivo de los servicios es almacenado en el direcrtorio /etc y se puede corregir fácilmente usando cualquier editor de texto.

Los servicios tales como telnet, ftp y TFTP son definidos en este archivo. Examinando este archivo, usted puede determinar en que números de puertos es que estos servicios están disponibles. El archivo exhibe la siguiente información (parcial), que variará dependiendo de sus requisitos particulares:

```
#
# Network services, Internet style
#
# WELL KNOWN PORT NUMBERS
#
rtmp            1/ddp           #Routing Table Maintenance Protocol
tcpmux          1/udp           # TCP Port Service Multiplexer
tcpmux          1/tcp           # TCP Port Service Multiplexer
qotd            17/udp          # Quote of the Day
qotd            17/tcp          # Quote of the Day
                                # Jon Postel <postel@isi.edu>
msp             18/udp          # Message Send Protocol
ftp-data        20/tcp          # File Transfer [Default Data]
ftp             21/udp          # File Transfer [Control]
ftp             21/tcp          # File Transfer [Control]
                                # Jon Postel <postel@isi.edu>
ssh             22/udp          # SSH Remote Login Protocol
ssh             22/tcp          # SSH Remote Login Protocol
telnet          23/tcp          # Telnet
```

```
smtp          25/udp      # Simple Mail Transfer
msg-auth       31/udp     # MSG Authentication
dsp           33/udp      # Display Support Protocol
time          37/udp      # Time
                          # Christopher Leong <leong@kolmod.mlo.dec.com>
finger        79/udp      # Finger
http          80/udp      www www-http # World Wide Web HTTP
kerberos      88/tcp      # Kerberos
sgmp          153/tcp     # SGMP
irc           194/tcp     # Internet Relay Chat Protocol
at-8          208/tcp     # AppleTalk Unused
ldap          389/tcp     # Lightweight Directory Access Protocol
tserver       450/tcp     # Computer Supported Telecomunication Applications
rcp           469/tcp     # Radio Control Protocol
pov-ray       494/tcp     # POV-Ray
login         513/tcp     # remote login a la telnet;
who           513/udp     # maintains data bases showing who's
shell         514/tcp     # cmd
syslog        514/udp     #
printer       515/udp     # spooler
talk          517/udp     # like tenex link, but across
socks         1080/udp    # Socks
tripwire      1169/udp    # TRIPWIRE
tripwire      1169/tcp    # TRIPWIRE
nessus        1241/udp    # nessus
nessus        1241/tcp    # nessus
```

***El Archivo inetd.conf***

El archivo inetd.conf es en almacenado en el directorio /etc y se puede corregir fácilmente usando un editor de texto. El demonio inetd maneja a los demonios de los servicios de red TCP/IP según la información contenida en el archivo inetd.conf. Para reconfigurar los procesos ejecutándose en inetd, edite el archivo inetd.conf y envíele una señal de SIGHUP al inetd.

Cada línea en el archivo contiene la información requerida para manejar a un demonio de red en particular. El archivo exhibe la siguiente información (parcial), la cual variará dependiendo de la personalización para satisfacer los requisitos particulares:

```
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#:INTERNAL: Internal services
#finger       stream      tcp     nowait      nobody      /usr/libexec/tcpd   fingerd -s
#ftp          stream      tcp     nowait      root        /usr/libexec/tcpd   ftpd -l
#login        stream      tcp     nowait      root        /usr/libexec/tcpd   rlogind
#nntp         stream      tcp     nowait      usenet      /usr/libexec/tcpd   nntpd
#ntalk        dgram       udp     wait        root        /usr/libexec/tcpd   ntalkd
#shell        stream      tcp     nowait      root        /usr/libexec/tcpd   rshd
#telnet       stream      tcp     nowait      root        /usr/libexec/tcpd   telnetd
#uucpd        stream      tcp     nowait      root        /usr/libexec/tcpd   uucpd
#comsat       dgram       udp     wait        root        /usr/libexec/tcpd   comsat
#tftp         dgram       udp     wait        nobody      /usr/libexec/tcpd   tftpd /private/tftpboot
```

```
#bootps     dgram     udp   wait      root      /usr/libexec/tcpd   bootpd
```

***Control de Mensaje de Protocolos de Internet (ICMP)***

El ICMP es el protocolo de la resolución de problemas de TCP/IP y es una parte requerida de la pila del TCP/IP. El ICMP se específica en RFC 792. Este permite que los host y las pasarelas/gateways del Internet reporten los errores a través de mensajes ICMP. Estos mensajes son encapsulados en la porción de datos del paquete IP y en última instancia son orientados al módulo de software del IP en el sistema de origen.

Cuando un datagrama viaja hacia su destino tendrá que atravesar una serie de pasarelas (routers), las cuales procesan el datagrama para dirigirlo adecuadamente hacia su destino. Si una pasarela no puede encaminar ese datagrama, o bien detecta alguna condición especial en la que se ve incapacitada para hacerlo (congestión de red, líneas fuera de servicio, etc.), entonces ese datagrama se pierde.

Estas y otras circunstancias en el tratamiento de los datagramas en su viaje hacia el destino, hacen necesario la creación de un mecanismo que, al menos, informe de estas situaciones al host origen, para que sea consciente de los problemas que ha sufrido el datagrama que ha enviado y, si procede, tome las acciones oportunas. De aquí nace el protocolo ICMP.

El protocolo ICMP (Internet Control Message Protocol), es un mecanismo que informa de la aparición de errores en la manipulación de los datagramas. Siempre que una pasarela detecte un error o excepción en un datagrama, utiliza el protocolo ICMP para informar al host origen de la circunstancia. ICMP no realiza ninguna acción para corregir el error que se haya producido, solamente se encarga de comunicarlo al host origen para que éste realice las acciones oportunas para corregir el error.

Originalmente, ICMP fué diseñado como un protocolo para los routers, sin embargo los host también lo pueden utilizar. Los mensajes ICMP van encapsulados en datagramas IP. El destino del mensaje ICMP no será la aplicación del usuario en el host destino, sino que deberá ser interpretarlo por su módulo ICMP. Si un mensaje ICMP afecta a una aplicación del usuario, ICMP deberá articular los mecanismos necesarios para comunicar a la aplicación el evento ocurrido.

Aunque este protocolo fué diseñado para detectar las incidencias que se producen en el transporte de un datagrama hacia el host destino, no todas ellas pueden ser detectadas. Entre estas causas se encuentra la pérdida de un datagrama que lleva un mensaje ICMP. En este punto, podríamos pensar que para solucionar este problema, esta pérdida podría ser notificada con otro mensaje ICMP. Más que solucionar el problema, lo estaríamos agravando cuando la razón de esa pérdida sea una congestión en la red. Por eso, NO SE PERMITE la notificación de mensajes ICMP causados por la pérdida de datagramas que lleven un mensaje ICMP. Otra norma general que impone este protocolo es que las notificaciones de error se envían SÓLAMENTE al host origen.

| Campo | Tipo de mensaje |
|---|---|
| 0 | Echo Reply |
| 3 | Destination Unreachable |
| 4 | Source Quench |
| 5 | Redirect |
| 8 | Echo Request |
| 11 | Time exceeded |
| 12 | Parameter Problem |
| 13 | Timestamp Request |

| | |
|---|---|
| 14 | Timestamp Reply |
| 15 | Information Request |
| 16 | Informatión Reply |
| 17 | Address Mask Request |
| 18 | Address Mask Reply |

Cada tipo de mensaje ICMP tiene su propio formato, aunque todos ellos comienzan con tres campos comunes, el resto puede variar en función del tipo de mensaje: El campo TIPO identifica el tipo de mensaje ICMP (ocupa 8 bits). En la Tabla anterior se muestran los distintos tipos de mensajes que contempla este protocolo. El campo CÓDIGO se usa para dar más información acerca del tipo de mensaje ICMP (8 bits). Y el último campo, contendrá el CHECKSUM de todo el mensaje ICMP (16 bits). El cálculo del checksum es el mismo que en IP, solo que en este caso cubre todo el mensaje ICMP.

En la tabla se muestran los distintos tipos de mensajes ICMP que considera el protocolo. De todos ellos, los más importantes son los seis primeros, que a continuación pasamos a describir brevemente:

- **Echo Request y Echo Reply:**
  Son dos mensajes que se usan conjuntamente para determinar la alcanzabilidad de un host o una pasarela. Normalmente son utilizados por los hosts, de forma que estos pueden extraer información acerca del estado del host remoto, el retardo que introduce la red en la entrega de los mensajes y el porcentaje de mensajes perdidos.
- **Destino inalcanzable (Destination unreachable):**
  Se trata de un mensaje que es generado por una pasarela cuando no puede encaminar un datagrama. Existen diferentes causas que provocan la emisión de este mensaje y que están codificadas en el campo código del mensaje ICMP. El mensaje es dirigido al host que ha enviado el datagrama, y en su interior se especifican los primeros 64 bits del datagrama que lo ha causado.
- **Source Quench:**
  Es un mensaje que utilizan las routers para frenar el ritmo de inyección de mensajes en la red de un determinado host. Esta situación se produce cuando una pasarela se ve sobrecargada con la recepción de datagramas (una posible situación de congestión) teniendo que descartar algunos por falta de buffers. Cuando se produce esta situación, la pasarela envía un mensaje de este tipo al host origen del datagrama descartado, diciéndole que baje el ritmo de inyección de datagramas ya que en ese momento hay una situación temporal de congestión.
- **Cambio de ruta (redirect):**
  Este mensaje es utilizado por una pasarela para indicar a un host de su red IP, un cambio en su tabla de encaminamiento, debido a la existencia de otra pasarela en la red que es más idónea que la que está utilizando actualmente.
- **Tiempo de vida agotado (Time Exceeded):**
  Cuando una pasarela encamina un datagrama, una de sus tareas es decrementar en una unidad el campo TTL de la cabecera del mismo. Si tras la operación el campo vale “0”, debe descartar el datagrama y enviar un mensaje ICMP de este tipo hacia el host origen.

***Otras Fuentes de Información***

A continuación se les presenta una lista de fuentes importantes de información de la resolución de problemas:

- Manuales de cursos que haya tomado
- Otros libros
- RFCs
- Páginas web
- Grupos de Noticias

## ESCENARIOS DE RESOLUCION DE PROBLEMAS

Hay algunos problemas que usted va a enfrentar más a menudo que otros:

- Dificultad de poner hardware a funcionar correcta
- Problemas de resolución de nombre
- Conexiones intermitente entre los hosts
- Ninguna conexión entre ciertos host

En esta sección se incluyen procedimientos que usted puede seguir al tratar de solucionar problemas de averías problemas de la red. Discutiremos los siguientes problemas de red:

- Acabo de Instalar un Nuevo NIC, Pero No Puedo Interconectar
- Sólo Puedo Interconectar Usando Direcciones IP
- ¿Por Qué Son Algunos Host en Mi Red Inalcanzables?
- Puedo Interconectarme, Pero Pierdo 50 Por Ciento De Mis Paquetes
- ¿Por qué Consigo Conexiones De Red Intermitentes?

***Acabo de Instalar un Nuevo NIC, Pero No Puedo Interconectar***

Hay muchas razones para la ocurrencia de un problema tal como este, el más común implica fallas con el hardware y configuraciones de los módulos. Antes de involucrarnos en cuestiones demasiadas técnicas, uno debe hacer algún tipo de examinación básica para asegurarse de que el problema no es realmente un descuido. Si después de dicha examinación, se determina que el problema proviene de situaciones más generales, puede ser necesario hacer el ajuste a la configuración de la tarjeta.

***Asegúrese de que Otros Medios de la Red Estén Trabajando Correctamente***

Lo primero que uno debe de asegurarse cuando una nueva tarjeta de red no esté trabajando correctamente es si está funcionando correctamente el resto de los componentes de la red. No tiene ningún sentido pasar horas haciendo ajustes a la configuración del NIC cuando el problema podría ser fácilmente un cable malo de la red. Problemas similares podían también presentarse de un hub que está funcionando incorrectamente. Una solución a esta problematica es intentar conectar el cable en la NIC en una máquina que uno está seguro que trabaja correctamente. Si esa máquina experimenta problemas de la funcionalidad de la red similares a los de la máquina que contiene el nuevo NIC, entonces es cierto que el problema yace en otra parte en la red. Si usted no tiene otra máquina disponible para realizar una prueba de esta naturaleza, intente cambiar cualesquiera cables o dispositivo de red en su red.

***Asegúrese de Que el NIC Esté Asentado Correctamente***

No importa que en particular que también este configurada una tarjeta si no está asentada bien físicamente. Por lo General, esto significa que la tarjeta no está dentro del slot de la placa madre. Cuando esto sucede, el sistema no puede comunicarse con el NIC. Cuando la tarjeta está instalada, usted debe cerciorarse de que la tarjeta ajuste con seguridad en lugar. Uno nunca debe forzar una tarjeta a que ajuste; sin embargo, una fuerza moderada es a veces necesaria.

Si nada de esto funciona y aún esta sospechoso, a este punto, el usuario puede también intentar instalar el NIC en otro slot o ranura ya que puede ser que la original este en falla.

***Compruebe las Configuraciones de Red***

Si la tarjeta está bien instalada o probada en otro ordenador y funciona debemos pasar a usar un programa de la configuración, tal como netconfig o ifconfig, para asegurarnos de que los parámetros de red estén establecidos correctamente en el sistema. Estas utilidades de configuración de red, a través de una serie de preguntas, le permiten al usuario incorporar tal información como el nombre de host, la dirección IP de la máquina, el nombre del servidor y la pasarela (gateway) por defecto. Asegúrese de que toda esta infor-

mación se digite correctamente. Si esta información no está disponible, deberá obttenerla con el ISP o el administrador del sistema.

***Compruebe Las Configuraciones De Hardware***

Si se ha comprobado que no hay problema con el resto de la red y que el NIC está asentado correctamente, usted puede asumir con relativa confianza que la solución ya está cerca haciendo algunos ajustes a la configuración del NIC. Antes de hacer estos ajustes, sin embargo, compruebe para asegurarse de que la tarjeta esté siendo reconocida por la máquina. La manera más fácil de hacer esto es mirando los mensajes almacenados en el kernel ring buffer. En cada arranque del sistema, el kernel ring buffer registra cada mensaje del sistema. Por lo tanto, este almacenador intermediario (buffer) debe contener la información sobre el NIC, si es que fué detectado a la hora de cargar. El comando dmesg está específicamente diseñado para exhibir la información en el kernel ring buffer. Una importante observación es que una vez se llene este almacenador intermediario, continúa agregando los mensajes más recientes y eliminando los más viejos. Por esta razón, el usuario debe utilizar el dmesg cuanto antes después del arranque.

Al revisar la salida del dmesg, busque las palabras claves como eth0, marca del NIC, o cualquier otra información que se pueda relacionar con la tarjetas de la red. En esta etapa, la principal preocupación es ver si la información está presente más que intentar de interpretarla. Si está presente, el usuario sabe que la tarjeta está siendo reconocida por el sistema. Si no está presente, la tarjeta no se ha detectado. El paso siguiente es agregar el módulo apropiado para el NIC. Para este propósito, es mejor utilizar una utilidad de la configuración, tal como netconfig, si es que está disponible. Las utilidades de la configuración permiten que usted realice fácilmente cambios a las configuraciones de hardware. Si el usuario aún experimenta problemas después de intentar una utilidad de configuración, necesitará aplicar alguna de la configuración manual que discutiremis más adelante en esta sección.

Si no hay una utilidad de configuración disponible en el sistema, los módulos se pueden cargar usando el comando del modprobe. Si se asume que depmod -a ha sido ejecutado desde el último build (etapa de compilación del kernel que prepara los módulos para luego poder ser linkeados) y la instalación más reciente del kernel completo o de un módulo nuevo, el modprobe consultará el árbol de dependencia en el archivo modules.conf (o en versiones anteriores, conf.modules) para confirmar qué serie de comandos necesita para cargar manualmente un módulo en el kernel. El modprobe también permite que el usuario especifique las direcciones IRQ y de I/O que se deben utilizarse para su hardware. Por ejemplo, para cargar un módulo para un NIC NE2000 de nombre ne2 en la dirección IRQ 15 y una dirección base de I/O 0x380, usted ejecutaría el modprobe con el sintaxis siguiente:

***# modprobe ne2 irq=15 ioprot=0x380***

Utilice el modprobe para cargar el módulo necesitado por su NIC usando el mismo sintaxis y los parámetros específicos a su tarjeta. Si no estea seguro de los ajustes de las direcciones IRQ y I/O, lo discutiremos más adelante en esta sección. Después de reiniciar el sistema, compruebe los mensajes de sistema usando el comando dmesg. Si todavía no hay información pertinente a su NIC, puede ser necesario que intente cargar un módulo diferente. Si aparece la información, está claro que el módulo funcionó y que fué cargado, y el sistema reconoce que el NIC está instalado.

***Compruebe la Dirección de IRQ y la Dirección Base de I/O***

Si el sistema reconoce el NIC, pero aún no hay comunicación con la red, puede ser que el NIC esté utilizando una dirección de IRQ o una dirección base de I/O que esté en conflicto con la de otro dispositivo. Antes de intentar configurar estos aspectos del NIC, usted debe entender que significan estos términos y como ellos trabajan.

IRQ son la siglas para la petición de la interrupción. Un ajuste de IRQ le dice a una computadora dónde

buscar señales de interrupción de un dispositivo instalado. Por ejemplo, cuando la tarjeta de red realiza una operación, enviará una señal al computadora para que esta sepa su estado. La computadora entonces sabrá como asignar prioridades a sus procesos. Si más de un dispositivo es asignado una sola dirección IRQ, el computador se confundirá en de donde proceden las señales. Por lo tanto, usted no puede asignar a múltiples dispositivos la misma dirreción IRQ.

Una lista de qué direcciones IRQ están en uso por un sistema y los dispositivos que las usan está situada en el archivo /proc/interrupts. Un usuario puede deducir al ver este archivo si el NIC está presente en cualquier IRQ en particular o uno puede buscar IRQs disponible para utilizar al hacer las especificaciones para el módulo del NIC. Aquí le mostramos un archivo /proc/interrupts:

```
k1k1@kikla:~$ cat /proc/interrupts
           CPU0
  0:    5473807    XT-PIC  timer
  1:       3982    XT-PIC  i8042
  2:          0    XT-PIC  cascade
  3:      38394    XT-PIC  orinoco_cs
  5:         52    XT-PIC  ALI 5451
  7:         11    XT-PIC  parport0
  8:          1    XT-PIC  rtc
  9:      15194    XT-PIC  acpi
 10:        101    XT-PIC  eth0
 11:          7    XT-PIC  yenta
 12:     298498    XT-PIC  i8042
 14:      24523    XT-PIC  ide0
 15:      49204    XT-PIC  ide1
NMI:          0
LOC:          0
ERR:          0
MIS:          0
```

Aquí, la primera columna es la dirección de IRQ, y la última columna indica qué dispositivo está usando una dirección particular.

Una dirección de I/O específica el rango de direcciones en memoria que puede ser utilizada por un dispositivo instalado a un sistema. Como un ajuste de IRQ, un dispositivo que es asignado una dirección I/O no puede estar en conflicto con otros dispositivos o el dispositivo no funcionará, y en algunos casos, el sistema cesará de funcionar en su totalidad. Para ver una lista de las direcciones de I/O en uso y los dispositivos que los están utilizando, visualice el archivo /proc/ioports. Al igual que /proc/interrupts, este archivo es útil para determinar si el NIC está trabajando o para encontrar un ajuste disponible si no está trabajando. Debajo está la muestra de un archivo /proc/ioports:

```
k1k1@kikla:~$ cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
0040-005f : timer
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0100-013f : pcmcia_socket0
0170-0177 : ide1
01f0-01f7 : ide0
```

***02f8-02ff : serial***
***0370-0371 : pnp 00:00***
***0376-0376 : ide1***
***0378-037a : parport0***

Aquí uno puede ver un listado de las direcciones I/O seguida inmediatamente por los nombres de los dispositivos conectados.

Si, después de realizar los procedimientos descritos en esta sección, el NIC aún no funciona como debe, contacte al vendedor o al fabricante del NIC para la ayuda adicional.

## Puedo Conectarme Solamente Con Direcciones IP

Este problema no involucra los cables de la red puesto que si hay la conexión usando la Dirección IP del servidor. El único problema aquí está en resolver los nombres simbólicos a las direcciones IP, que se asocian en DNS. Los siguientes son algunos posibles errores que pueden ocurrir y la solución sugerida.

***El Archivo de Configuración /etc/host.conf Pueden que Contenga Errores de Sintaxis***

El administrador de sistema necesita estar seguro que el orden en este archivo es la exacta en la cual resolver nombres de host a direcciones IP. Si se incluyen opciones en el archivo/etc/host.conf, deben ocurrir en líneas separadas.

***order hosts nis bind***
***multi on***

En este ejemplo, la opción “order” le dice al sistema que para determinar la información requerida primero revise el archivo /etc/hosts, después uso NIS para la resolución de nombres y finalmente use el BIND para consultar un servidor del DNS. La opción “multi on” le dice al sistema que se utilizarán direcciones múltiples.

***El Nombre Dominio y la Dirección IP del Servidor de Nombre se Pudieron Haber Establecido Incorrectamente***

El administrador de sistema necesita revisar el archivo /etc/resolv.conf en el host local para verificar que el sintaxis está correcto. Si el sintaxis no está correcto en una línea en archivo /etc/resolv.conf, entonces el DNS simplemente ignora esa línea. Esto podría causar que el Servidor de Nombre en particular sea inasesible al DNS.

***Puede Ser Que el Servidor de Nombre No Responda***

El administrador de sistema debe tener por lo menos dos Servidores de Nombres en el archivo /etc/resolv.conf para asegurarse de que el segundo Servidor de Nombres se pueda utilizar para resolver nombres de host en el acontecimiento que el primer servidor de nombres no esté respondiendo.

***Puede Que el DNS en el Servidor de Nombres No Se Este Ejecutando***

Primero, el administrador del sistema necesita comprobar si el archivo de boot (cargar) del named o si está presente el archivo /etc/named.boot. El demonio named debe estar ejecutándose en el Servidor de Nombres para que el DNS pueda estar operacional.

### El Servidor del DNS No está funcionando Correctamente

La manera más fácil de confirmar que el servidor del DNS está funcionando correctamente es usando el comando del nslookup. El administrador de sistema necesita cerciorarse de que el comando nslookup asigna el servidor de nombre por defecto a la dirección IP. La salida del comando nslookup es mostrado a continuación para el servidor de nombre ficticio “ns.codigolibre.org”:

***$nslookup***
Default Server: ns.codigolibre.org

```
Address:        10.0.0.1
   >set all
Default server:         ns.codigolibre.org
Address:        10.0.0.1

Set options:
  novc                  nodebug         nod2
  search                recurse
  timeout = 0           retry = 2       port = 53
  querytype = A         class = IN
  srchlist = sede.codigolibre.org"
```

El administrador de sistema puede utilizar el comando "set all" después del prompt para obtener el dominio por defecto, lista de búsqueda, el tipo de consulta, y otras opciones además de la dirección del servidor de nombre, según lo mostrado en el ejemplo anterior.

Si el administrador de sistema no puede conseguir esta salida, él o ella sabe que hay un problema con el servidor de nombre y verifica otra vez el archivo /etc/resolv.conf.

Una vez que el nslookup esta funcionando y dá la salida deseada, el administrador de sistema no debe tener dificultad al conectarse con el servidor usando nombres de host.

Para los host más comúnmente usados, el archivo /etc/hosts puede ser utilizado para la traducción de los nombres del dominio a las direcciones de IP y de los nombres cortos y viceversa.

***¿Por qué son inalcanzables algunos host en mi red?***

Este problema puede venir en varias formas extendiendose desde influencias globales a las locales. Las posibilidades son relativamente fáciles de apuntar una vez que se entienden cuales herramientas serán necesitadas diagnosticar el problema. Este topico de resolución de problemas se divide en tres secciones. Primero, las influencias globales, las cuales suelen ser la fuente del problema la mayor parte del tiempo, son discutidos desde la perspectiva del mundo exterior (nonuser). En segundo lugar, el factor local que afecta la conexión de una red con un host. La influencia local es basada en la localización inmediata. Las causas obvias y básicas del fallo de la red se cubren en esta sección. Finalmente, la tercera sección proporciona diagnóstico si las soluciones proporcionadas fallan. A continuación se muestra un panorama específico.

La idea de una LAN es conectarse con un host, un Web site, o una computadora usando una red 10BaseT. La conexión de red va directo a un hub, que, alternadamente, entra en un zócalo de pared RJ-45 (la topología de la red no es completamente reelevante para esta parte). La estacion de trabajo se sitúa en una pequeña oficina con varias otras computadoras que funcionan con una interface 10BaseT.

## Influencias Globales

Las influencias globales son factores que los usuarios no pueden controlar. Los problemas pueden implicar:

- El host debe apagarse en el proceso de reinicio del sistema.

  Los administradores de sistema necesitan actualizar constantemente sus computadoras con un mejor hardware y software. Este proceso colocará generalmente al ordenador fuera de línea temporalmente. En este caso, no tiene nada hacer con la configuración del 10BaseT. También, un host puede tener períodos donde

se está comunicando con varias computadoras, asi, de esta manera la red esta ocupada. Por esta razón, una conexión pudo fallar.

***Solución***

La solución más fácil es esperar e intentar más adelante. Una herramienta de diagnóstico simple y con muchos recursos que pueden ser usados es el comando ping. Corriendo este comando con la dirección IP, un usuario puede descubrir rápidamente si el ordenador está corrriendo basado en losreportes de paquetes perdidos.

- Los operadores de red pudieron no haber colocado o haber colocado incorrectamente su dominio con el Domain Name System (DNS).

  Esto evitaría que todos los usuarios en el Internet tuvieran acceso al Web site o al host.

***Solución***

Utilice un newsgroup del usuario para encontrar información sobre este host o para hacer una búsqueda más detallada para el nombre/URL del host con un motor de búsqueda. Esto puede revelar la información adicional al dominio apropiado.

- Otro ejemplo de no poder tener acceso a un host es cuando ha ocurrido una interrupción física en la conexión.

  Se ha interrumpido el cable que víncula el host particular al Internet. Una sección del cable se pudo haber desconectado o peor aún cortada, preveniendo al cables de enviar y recibir los paquetes de la transmisión.

***Solución***

La única opción es esperar hasta que se reparan los cables.

- Al comunicarse por el Internet, se necesitan un conjunto de protocolos para que asi se puedan comunicar las computadoras entre si. Pasarelas son utilizadas para transferir información de un protocolo a otro.

  Diferentes redes pueden usar diferentes protocolos tal como TCP/IP o IPX/SPX. Siempre y cuando estos protocolos pasan a través de una pasarela que víncule los dos protocolos, la comunicación prevalecerá. Si la pasarela sale fuera de línea, entonces reinará la posibilidad de ninguna comunicación al mundo exterior.

***Solución***

Si la pasarela está fuera, esperar que esta sea corregida es la única solución.

## Influencias Locales

Las influencias locales implican factores que el usuario tiene poco control, incluyendo:

- Una conexión física de un cable de UTP a una tarjeta de Ethernet se desconecta o se está conectando mal.

  La solución es comprobar si la tarjeta de Ethernet está trabajando correctamente y si está funcionando con los ajustes correctos. Otra cosa es que los LED en el NIC deben estar encendiendo si hay actividad.

- El medio de conexión (es decir, twisted pair) se ha dañado entre el terminal del usuario y el hub.

Esto ocurre a menudo cuando los cables no se construyen correctamente. Cuando los cables están a la interperie, la posibilidad de que se mal traten y de que los alambres se rompan dentro del protector de revestimiento aumenta. La solución es absolutamente simple: los alambres deben ser colocados donde la gente no puedan caminar sobre ellos, rodar sillas sobre ellos, o colocar cualquier cosas sobre ellos.

## Diagnóstico

Si las soluciones presentadas anteriormente no resuelven el problema, pruebe con las siguientes:

- Asegúrese de escribir la URL correctamente. Por ejemplo puede ser que escriba la letra x por la letra c al escribir “com”.
- ¿Es este host legítimo? Compruebe para ver si el host tiene una dirección inválida comprobando con el DNS.
- Ejecute un ping con la Dirección IP del host para ver si hay una respuesta. El comando es así:
  ***$ ping 10.0.0.105 -c 20***
- Si tiene problemas al conectar un ordenador en el hogar, compruebe el cuarto punto de Influencias Globales y la primera bajo Influencias Locales. Los protocolos se pueden pasar por alto al instalar el sistema operativo.

***Puedo Salir, Pero Pierdo 50 Por Ciento De Mis Paquetes***

Un paquete es la estructura estándar que se utiliza para encapsular los datos que son enviados sobre una red TCP/IP. Cuando una computadora envía paquetes a una computadora en el mismo LAN, la computadora destinataria recibe los paquetes directamente. Cuando las computadoras no están en la misma red física, los paquetes se pasan a través de dispositivos de red hasta que alcanzan el sistema remoto. Para proporcionar servicio confiable, el TCP utiliza un protocolo basado en reconocimiento. Siempre que se envíe un paquete, la computadora de recepción envía un mensaje que indica que recibieron el paquete y que más paquetes deben ser enviados. Se utilizan números de serie para mantener los paquetes orden. El TCP normalmente envía un número de paquetes juntos y espera un solo reconocimiento por esos paquetes. Si el reconocimiento no se recibe, la computadora que envía vuelve a enviar esos paquetes y espera otra respuesta. Si un reconocimiento no se recibe otra vez, los paquetes se consideran perdidos.

Los ruido en la líneas y relámpago son dos tipos de interferencia que pueden contribuir a la pérdida de paquetes; sin embargo, interferencia física no es una causa significativa de la pérdida de paquetes. Normalmene los paquetes son perdidos debido a congestión en la red. Pueden ser retrasados dentro de bucles de enrutamiento, ser recibidos fuera de orden o desechados debido a desbordamiento. Durante el volumen de tráfico alto, puede ser que un sistema reciba más paquetes que los que puede procesar. Una vez que los buffers esten llenos, un desbordamiento de los buffers puede ocurrir, lo que significa que desecha cualquier paquete nuevo hasta que el el buffer ya no este lleno. El desbordamiento del buffer es la razón más común de la pérdida de paquetes y puede ocurrir en el dispositivo físico, el driver del dispositivo, o la puesta en práctica del protocolo. Siempre que un paquete sea desechado debido al desbordamiento del buffer, la computadora de recepción no podrá reconocer el sistema de paquetes. La computadora que envía medirá el tiempo de entrega, declarará los paquetes perdidos, y enviará ese conjunto de paquetes otra vez. Alta pérdida de paquetes aumenta la congestión en la red y causa perdida de efectividad en la red.

Determinación la fuente de la pérdida de paquetes puede ayudar en aislar problemas dentro de una red. La pérdida de paquetes se puede detectar con un número de utilitarios estándares.

***El Utilitario ping***

Ping es un programa utilitario básico usado para comprobar si hay una respuesta en la red de un dispositivo. Ese envía una petición de eco ICMP al sistema remoto y espera por una respuesta por eco del ICMP. Los mensajes del ICMP funcionan en la capa de red, así que ping recibirá una respuesta de una máquina remota sin importar la plataforma que la máquina está funcionando. El ping le mostrará el tiempo de reacción al host, el TTL del paquete y un número de secuencia. El campo TTL realmente no es un campo de tiempo, sino una cuenta de los saltos que disminuye de un número base, normalmente 255, cada vez que el paquete encuentra un enrutador o una pasarela en su trayectoria. Una vez el contador llega a 0, el paquete es desechado.

Una vez el comando ping se termina (usando la combinación de teclas CONTROL+C), se efectuá un resumen de las respuestas y el porcentaje de la pérdida de paquetes se exhibe. Así, que podemos usar el ping para demostrar si está ocurriendo una alta pérdida de paquetes, ping tiene tres opciones las cuales pueden ser útil:

**-v** Cambia al modo verbose. Imprimen todos los mensajes del ICMP, no solamente respuestas del eco.

**-c** Esta opción es la cuenta con un argumento numérico, ping enviará solamente la cantidad de paquetes dados en el argumento.

**-f** Esto es el ping de inundación, una opción que pone rápidamente mucha carga en la red. El ping puede rápida y repetidamente efectuar ping a otros hosts (esta opción requiere privilegios del superuser).

Los problemas, tales como el reporte falso de porcentaje de la pérdida de paquetes, especialmente 50 por ciento, existe con algunos clientes de ping más viejos. Para solucionar este problema, simplemente debe actualizar el paquete de las herramientas de red “net-tools” a la última versión disponible para su distro.

***El Utilitario netstat***

El comando “netstat -i” mostrará la información sobre las interfaces actualmente configuradas en la máquina local. La salida puede exhibir los paquetes pérdidos o los problemas de la máquina local con los paquetes duplicados o dañados. La siguiente salida se abrevia para demostrar las columnas importantes:

```
# netstat -i
Kernel Interface table
Iface   MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0        0      0      0      0       109      0      0      0 BMU
eth1   1500   0        0      0      0      0         0    102    102      0 BMRU
eth2   1500   0     4575      0      0      0      2572      2      0      0 BMRU
lo    16436   0     1553      0      0      0      1553      0      0      0 LRU
sit0   1480   0        0      0      0      0         0    100      0      0 ORU
```

Las columnas RX son para los paquetes recibidos y las columnas TX son para los paquetes transmitidos. En el ejemplo anterior, ningunos paquetes dañados (ERR), los paquetes caídos (DRP) o las pérdidas debido al overrun (OVR) han ocurrido. Si ocurren paquetes dañados, es probable que es un problema de hardware exis en la red. Si los desbordamientos no están ocurriendo en la máquina local, el problema puede estar ocurriendo en otra parte en la red.

***El Utilitario traceroute***

La herramienta traceroute se puede utilizar para aislar los problemas de enrutadores en una red. El traceroute utiliza un mensajes pequeño UDP para buscar la pista y mapear la ruta que el paquete llevaría para arrivar a un host y el valor aproximado del tiempo del viaje. Usando la información obtenida de traceroute, uno puede continuar buscando el problema efectuando ping a los enrutadores en toda la ruta, lo que puede aislar el problema a un router en específico. La salida de traceroute mostrará nombres de host o si usa-

mos la opción -n, sólo mostrará las direcciones IP. Típicamente el comando traceroute se ejecuta de la siguiente forma:

***# /usr/sbin/traceroute***

### ¿Por Qué Logro Conexiones De Red Intermitentes?

Este problema se presenta muy probablemente por el hecho de que alguien está compartiendo una misma Dirección IP. Si las Direcciones IP de los usuarios se asignan estáticamente, entonces debemos cerciorarnos de utilizar la dirección que se nos fué asignada. Ingresar al sistema bajo una Dirección IP incorrecta causará problemas para el usuario y con la persona que está compartiendo la dirección.

#### *¿Cómo Puedo Detectar Si Alguien Está Utilizando Mi Dirección IP?*

Simplemente apague su computadora, inicie una sesión en otra máquina y efectué un ping al IP de su ordenador asignado (o sea al que acabas de apagar). Si se responde el ping, entonces alguien está usando esa dirección.

#### *¿Qué Debo Hacer Sobre Esto Situación?*

Lo primero que puede hacer es ejecutar un traceroute al otro host. Con esto determinará si el culpable está en la misma subnet que usted. Si no lo está, entonces el traceroute determinará la pasarela más cercana a la máquina que está usando el IP. Esta información asistirá al administrador del sistema en intentar corregir el problema. Las instrucciones para usar el comando traceroute la encontrará en las páginas man de su GNU/Linux.

Si el que está usando el IP está en la misma subnet, entonces podemos usar el comando arp para determinar la Dirección MAC ADDRESS de la otra máquina. Ya habiendo efectuado el traceroute, el otro host ahora ya está en la tabla del ARP de la máquina local. Usando el "arp -a" para enumerar el contenido de la tabla del ARP y podemos comprobar la salida para saber el MAC ADDRESS del culpable. Éste es un número que identifica de la tarjeta de red individual de la máquina y dependiendo de la topología de la red, podría dar al administrador bastante información para encontrar el terminal usando la Dirección IP ajena. Claro está, esto sólo trabajará si el usuario y el host usando la IP indebidamente están en la misma red.

## MONITOREAR EL RENDIMIENTO

Mientras mejor usted conoce su red, más fácil es encontrar los problemas que se le presenten en ella. El monitoreo y supervisión regular del rendimiento de su red le pueden proporcionar la detección preventiva de los problema que pueden surgir y agilizar el proceso de los esfuerzos para la resolución de los problemas futuros. En Esta sección se describen los elementos del monitoreo y la supervisión del rendimiento en como se aplican al proceso de resolución de problemas.

Los siguientes tópicos serán cubiertos en esta sección:

- Factores del Rendimiento
- Identificar la Degradación del Rendimiento
- Ambiente del Sistema
- Ambiente de la Red
- Aplicaciones Cliente/Servidor

### Factores del Funcionamiento

La identificación oportuna de entaponamientos es un factor dominante para mantener un constante nivel de rendimiento. La mayoría de los usuarios no son tolerantes de las aplicaciones cliente/servidor que su rendimiento sea inconsitente. Asi que el primer objetivo es obtener consistencia tanto en el fun-

cionamiento de las aplicaciones como el rendimiento. Claro se supone que para detectar inconsistencias, un administrador debe tener un parámetro contra el cual medir, la llamaremos la línea base del rendimiento que normal la red.

***Línea Base***

La línea base es grabar la actividad de la red en un momento de rendimiento satisfactorio para que nos sirva de parámetro para comparar contra futura actividad de la red. Las líneas base deben ser registradas cuando una red está funcionando correctamente. Si se introducen problemas a la red, el nuevo comportamiento de la red se puede comparar contra la línea base. Las líneas base se pueden utilizar para determinar entaponamiento, para identificar patrones tráfico denso y para analizar el uso diario de la red y los patrones de los protocolos usados en la red.

**Identificar la Degradación del Rendimiento**

Una aplicacióncliente/servidor de red que anteriormente funcionaba correctamente pero que recientemente se ha tornado inconsistente en su comportamiento típicamente indica que uno o más de sus recursos están siendo utilizados a sus límites.

***Sistema***

Puede ser que la aplicación este esperando en algunos recursos del sistema como por ejemplo:

- El CPU en el nodo parte del Cliente en el cual se ejecuta una parte de la aplicación.
- El CPU en el nodo parte del Servidor en el cual se ejecuta una parte de la aplicación.
- El CPU en el nodo del Servidor al que la aplicación necesita tener acceso (por ejemplo, un aplicación de tres capas que accesa una base de datos en el servidor).
- Memoria en el nodo del Cliente que la aplicación se ejecuta.
- Memoria en el nodo del Servidor que parte de la aplicación se ejecuta.
- Memoria en el nodo del Servidor que la aplicación necesita tener acceso

***Network***

Los recursos de la red que afectan el rendimiento de una aplicación incluyen:

- La LAN en la que el nodo Cliente está configurado
- La LAN (sistema operativo o aplicación) en la que el nodo del servidor está configurado.
- La LAN en la que la aplicación nodo del servidor está configurada.
- La WAN que conecta el nodo Cliente con el nodo Servidor (dependiendo de donde las aplicaciones se ejecutan con respecto a la red)
- Arquitectura y configuración de los dispositivos de comunicación.
- Protocolos usados entre el cliente y el servidor de la aplicación.

***Aplicaciones Cliente/Servidor***

Determinar el uso de los recursos del sistema y de red críticos no es suficiente. El diseño y la arquitectura de las aplicaciones cliente/servidor y el servidor de aplicacion es configurado y administrado son factores extremadamente importantes en análisis de funcionamiento. El funcionamiento de la aplicacion debe ser pobre por que la aplicacion de la arquitectura fueron diseñados mal. Varios factores deben ser considerados:

- ¿Es la mayoría de los datos accesados por los usuarios finales de solo lectura y lectura/escritura?•
- ¿Qué opciones el servidor de aplicaciones proporciona para maximizar funcionamiento en un sistema de un solo procesador?
- ¿Qué opciones el servidor aplicaciones proporciona para maximizar el funcionamiento en un sistema de múltiples procesadores?

- Es posible que los procesos del servidor de aplicaciones puedan estar ocupados pero el sistema esta siendo usado. ¿Cómo puede usted mejorar el uso del sistema con los procesos del servidor de aplicaciones?
- ¿Cuál es la arquitectura del servidor de aplicaciones? ¿Se centraliza o se descentraliza? ¿Por qué? ¿Es este arreglo constante con la arquitectura de red?

En el lado de la aplicacion, la arquitectura de aplicacion y cómo este código puede significativamente afectar la utilización de los recursos del sistema y de la red. ¿Por ejemplo, cómo es implementado el control de la version? ¿Hay un mecanismo que proporcione la información sobre el sistema de la carga y la red como resultado de los cambios realizados en la aplicacion?

***Establecer Pautas***

Para solucionar el problema de inconsistencia y pobre funcionamiento de la aplicacion, usted debe establecer pautas específicas en las áreas siguientes:

- Ambientes del Sistema
- Ambientes de la Red
- Aplicaciones Cliente/Servidores

***Ambiente de Sistema***

El ambiente de sistema incluye las dos áreas siguientes:

- Sistema del Hardware
- Sistema Operativo

***Sistema del Hardware***

Las aplicaciones cliente/servidor se ejecuta por lo menos en dos sistemas pero puede ser que funcione en más. Típicamente, tres sistemas críticos son asociados:

- PC del usuario (el interfaz GUI).
- El sistema que funciona como servidor de archivo para la PC del usuario. El ejecutable de la aplicacion que reside en el servidor de archivo del sistema.
- El host que esta configurado como el servidor.

La configuración de todos los sistemas debe estar claramente definida. Los elementos importantes de la configuración son los siguientes:

- **Procesador**
  - Tipo de CPU (Pentium de Intel, alfa Digital, sol SPARC)
  - Número de CPUs

  ¿Cuál es el número máximo de CPUs que puede ser instalado en este sistema?

- **Memoria**
  - **Memoria Real**

  ¿Cuánta memoria está instalada en el sistema? ¿Cuál es la cantidad máxima de memoria que podría ser instalada?
  - **Memoria Virtual**

  ¿Monto de memorias secundarias que han sido configuradas en el sistema operativo?
  ¿Cuál es la cantidad máxima de memoria que el sistema opetativo soporta?

- **Interfaz de la red**
  - ¿Qué tipo de interfaz de la red ha sido configurada en el sistema?
  - ¿Están la interfaz y el sistema operativo optimizados para transmitir la Unidad Máxima de Transmisión (MTU) definido por el protocolo en uso?

• ¿Están la interfaz de la red y el sistema operativo optimizados para aprovecharse del bús del sistema? ¿Por ejemplo, es el USB preferible a ISA o al PCI?

**• Disco**

• ¿Qué tipo de controladora y manejadora están instalado en el sistema?

En la siguiente tabla listamos los elementos del sistema y los factores que un administrador de la red debe tomar en consideración.

**Sistema Operativo**

Usted también debe determinar e indicar exactamente cuales componentes de las aplicaciones funcionan en sistemas Cliente/Servidor y cómo estos funcionan. Esta sección esplicara las herramientas para monitorear las herramienta para Linux.

En los sistemas GNU/Linux, usted puede utilizar los siguientes comandos para proporcionar información de como el sistema esta funcionando:

• vmstart
• ps
• uptime

***El Comando vmstat***

El comando vmstat (estadística de la memoria virtual) proporciona información de la memoria virtual, el acceso del disco, y el uso de la CPU. Los resultados proporcionados por este comando son promediados a partir del tiempo de que el sistema fué inicializado. Para obtener la información sobre la actividad máxima del sistema (que puede indicar entaponamiento potenciales del sistema), especifique un intervalo como un argummento del comando del vmstat.

| Elementos Del Sistema | Consideraciones del Administrador de la Red |
|---|---|
| Configuración del Sistema Cliente | Determinar el procesador en el sistema cliente: Intel, Spac, etc<br>Determinar la velocidad del reloj: 233 MHZ, etc. |
| Configuración del Sistema Operativo Servidor | Determinar el procesador en el sistema operativo servidor: Intel, Spac, etc<br>Determinar la velocidad del reloj: 233 MHZ, etc. |
| Configuración del Servidor de Applicación | Determinar el procesador en el servidor de aplicaciones del sistema: Intel, Spac, etc<br>Determinar la velocidad del reloj: 233 MHZ, etc. |
| Memoria Instalada/Memoria Maxima en el Sistem Cliente | Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras. |
| Memoria Instalada/Memoria Maxima en el Sistema Servidor | Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras. |
| Memoria Instalada/Memoria Maxima en el Servidor de Aplicaciones | Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras. |
| Memoria Virtual/Memoria Maxima Virtual en el Sistema Cliente | Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras. |
| Memoria Virtual/Memoria Maxima Virtual en el Sistema Servidor | Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras. |
| Interface de Red del sistema Cliente | Determinar las interfaces: Ethernet, Token Ring, y/o FDDI |
| Interface de Red en el Sistema Operativo del Servidor | Determinar las interfaces: Ethernet, Token Ring, y/o FDDI |
| Interface de Red en el Servidor de Aplicaciones | Determinar las interfaces: Ethernet, Token Ring, y/o FDDI |

Para la estadística de la CPU, especifique un intervalo de dos segundos. Para la estadística del disco, especifique un intervalo de 60 segundos. El comando vmstat duerme por el intervalo definido. Típicamente, si el tiempo libre de la CPU es mayor de 20 por ciento, implica que el sistema es límite de I/O o límite de la memoria. El tiempo libre de la CPU incluye la siguiente información sobre el tiempo de la CPU:

- La CPU no esta en uso por que no tiene nada hacer.
- La CPU está por esperando memoria.
- La CPU está esperando por una entrada/salida (I/O).

Examine la salida del comando vmstat en las columnas datos descrita en la siguiente tabla:

***El Comando ps***

Cuando el shell lanza un programa, se crea un nuevo proceso y se le asigna un número entero (PID) entre 1 y el 30,000, del cual se tiene la seguridad que va a ser unívoco mientras dure la sesión. Se puede verificar ejecutando el comando "ps", el cual muestra los procesos activos que tiene asociados a la terminal.

Un proceso que crea a otro se lo denomina proceso padre. El nuevo proceso, en este ámbito se le denomina proceso hijo. Este hereda casi la totalidad del entorno de su padre (variables, etc.), pero sólo puede modificar su entorno, y no el del padre.

La mayoría de las veces, un proceso padre se queda en espera de que el hijo termine, esto es lo que sucede cuando se lanza un comando, el proceso padre es el shell, que lanza un proceso hijo (el comando).

Cuando este comando acaba, el padre vuelve a tomar el control, y recibe un número entero donde recoge el código de retorno del hijo (0 = terminación sin errores, otro valor = aquí ha pasado algo).

Cada proceso posee también un número de "grupo de procesos". Procesos con el mismo número forman un solo grupo y cada terminal conectado en el sistema posee un solo grupo de procesos. (Comando ps -j) si uno de nuestros procesos no se halla en el grupo asociado al terminal, recibe el nombre de proceso en background (segundo plano).

| Column | Descripción |
|---|---|
| r | La columna provee información en jobs que están disponibles. Si el siguiente número es alto, esto implica que el CPU es forzado a cambiar entre jobs disponibles. Este número debe indicar que el sistema esta destinado al CPU.. |
| b | Esto provee innformacion en jobs durmiendo en una prioridad negativa, usualmente por un proceso que esta esperqando por disco, cinta, u otro recurso. Si este número es alto y el tiempo del CPU es alto, el sistema puede ser destinado a entrada y salida. |
| w | Esta columna específica el número de jobs ejecutados en los pasados 20 segundos y han sido sacado de la swap. Si este campo no es cero, el sistemas puede no tener suficiente memoria. |

Es posible utilizar algunas variantes del comando "ps" para ver qué procesos se tiene en el equipo:

**ps :** muestra el número de proceso (PID), el terminal, el tiempo en ejecución y el comando. Sólo informa de nuestra sesión.
**ps -e :** de todas las sesiones.
**ps -f :** full listing: da los números del PID, del Ppid (padre), uso del procesador y tiempo de comienzo.
**ps -j:** da el PGID (número de grupo de los procesos - coincide normalmente con el padre de todos ellos).

El comando del ps, con opciones como aux y ef, proporciona información útil de los procesos que se ejecutan en el sistema. Provee información en:
- Uso del CPU por proceso (ver la columna de %CPU)

• Uso de la memoria por proceso (ver la columna de %MEM)
• Estado actual del proceso (ver la columna STAT)

***El Comando uptime***

El comando uptime proporciona información útil acerca de:

• El total del tiempo que el sistema ha estado ejecutándose.
• El número de usuarios ingresados en el sistema.
• Los promedios de las cargas de trabajos activos en el sistema en los previo 1, 5, y 15 minutos.

El kernel mantiene información sobre el promedio del número de trabajos activos en el sistema en los últimos 1, 5, y 15 minutos. El primer número proporciona información sobre la carga actual del CPU. Si el número es mayor a 4, el sistema puede ser limitado por CPU.

Otra herramienta disponible bajo licencia también GPL, es el comando top, este exhibe y actualiza la información sobre los quince primeros procesos del sistema.

**Ambiente de Red**

Desde el punto de vista de la red, existen diferencias significativas entre ejecutar una aplicación cliente/servidor sobre un LAN y ejecutar la misma apliación sobre un WAN. El impacto de la latencia de la red es más pronunciado en una WAN. Sobre todo esta diferencia es porque muchos segmentos de una WAN son de 56 Kbps o 1.5 Mbps, mientras que la mayoría de los segmentos de la LAN son de 10 Mbps ó 100 Mbps.

Los retrasos debidos a propagación y los introducidos por el procesamiento de los paquetes de los enrutadores afectaran el rendimiento de la aplicación client/server. Estos retrasos pueden ser más pronunciados en WANs que en LANs.

***Factores del Rendimiento***

En el área de red, el rendimiento de una aplicación puede ser afectada por lo siguiente:

- Pila del Protocolo
- Arquitectura del Enrutamiento
  - Protocolo de Enturamiento
  - Configuración del Enrutador
  - Saltos del Enrutador
- Ambiente WAN
- Ambiente LAN

Para resumir, la información siguiente es esencial para determinar si la red está entaponada:

- Uso promedio de los segmentos de la LAN que conectan el cliente y servidor.
- El Uso pico de los sistemas clave tales como nodo del cliente, sistema operativo del servidor y nodo del servidor de aplicaciones.
- Distribución de la trama en segmentos claves del LAN y la WAN. Use esta información para determinar si la aplicación o el servidor de aplicaciones tiene cualquier parámetro u opciones que afecten tamaño del paquete.
- Los tipos de protocolos en segmentos claves del LAN y la WAN. ¿Cuáles protocolos ponen mayor carga en la red? ¿Hay una manera de optimizar la carga de la red examinando donde los sistemas están situados en la red?

**Aplicaciones Cliente/Servidor**

El desarrollo y la implementación de aplicaciones cliente/servidor sobre una red a menudo requieren de un administrador de red y un desarrollador de aplicaciones. Para determinar cómo las aplicaciones cliente/servidor se ejecutan, usted debe confrontar las siguientes áreas:

- Arquitectura de las aplicaciones en términos del sistemas y de las redes
- Arquitectura de las aplicaciones en términos de los módulos (pantallas, rutinas)
- Control de la versión
- Prueba

## ADMINISTRACIÓN DE LA RED Y SNMP

La necesidad imperante de la administración de la red y las herramientas como el SNMP, se ha presentado debido a la complejidad de las redes de hoy. A menudo, la solución de problemas puede ser muy mejorada usando el SNMP (Simple Network Management Protocol).

A medidas que las redes crecen en tamaño y se tornan más complejas, la tarea de manejar los sistemas en una red se dificultan más. En los últimos años, un número de productos de la administración de la red han aparecido basados en un protocolo del suite del Internet conocido como SNMP. Esta sección nos introduce a la administración de la red usando el protocolo del SNMP.

Los siguientes tópicos son discutidos en esta sección:

- La Necesidad de la Administración de la Red
- Utilitarios Tradicionales de la Administración de la Red
- Administración Moderna de la Red
- Componentes de SNMP
- Estructura de la Información de Administración (SMI)
- Administración la Información Base (MIB) Espacio de Nombre del Objeto
- SNMP
- Otros Protocolos de la Admininstración de la Red

### *La Necesidad de la Administración de la Red*

Causado por el crecimiento de la redes y debido a este crecimiento la dificultadad de sus administración de estas redes nació la motivación para el desarrollo de una herramienta estandarizada de administración de la red. El dianóstico de lsos problemas se torno más y más dificil. La necesidad de herramientas de administración de red TCP/IP se convirtió pronunciada a mediado de la decada de los años ochenta a medida que estos problemas comenzaron a utilizar más recursos de muchas compañías. Había una necesidad obvia, para un conjunto de reglas estandardizado que gobernaran la identificación y que suplan acciones automatizadas para las diversas situaciones comunes que se presentan en la red.

Las recomendaciones para un protocolo estándar en el año 1988 (RFC 1052) incluyeron:

- Que el SNMP es adoptado como un estándar internacional.
- Que el SNMP sea extendido y mejorado continuamente.

En el 1988, un protocolo llamado SNMP ya existiá pero no fué estandardizado y una variedad de otros protocolos también se encontraban en uso.

El RFC 1052 resumió la decisión del IAB (Internet Arquitecture Board) para adoptar el SNMP como un estándar en todo el Internet y para mejorar su especificaciones y para garantizar su mejoramiento continuado. La IAB también decidió que los protocolos OSI (CMIP/CMOT) sean adoptados en última instancia como

el reemplazo del SNMP. Sin embargo, como con algunos otros protocolos de la OSI, éstos han caído por el wayside, y el SNMP ahora es el estándar.

***Utilitarios Tradicionales de la Administración de la Red***

En el pasado las herramientas tradicionales de la administración de la red tendian a ser demasiado propietarias y dispersas para la solución de problemas y la prevención de las fallas que ocurren día a día en redes heterogéneas. Se necesitaba una herramienta específicas para cada trabajo y no existían la disponibilidad de soluciones integrada. Por lo tanto, las herramientas tradicionales de administración de red no eran lo suficientemente robustas para las redes grandes.

En las redes pequeñas, el conjunto básico de utilidades que provee normalmente los paquetes de software TCP/IP, es suficiente para administrarlas . Ya hemos mencionado los utilitarios principales del conjunto que se proveé con el paquete TCP/IP:

**ifconfig**

Se utiliza para comprobar o actualizar los parámetros de una interface de red en el sistema local. Usando el ifconfig, usted puede cambiar la Dirección IP de un interfaz, su mascara de subred o su dirección de broadcast (difusión). Usted puede detener un interfaz de acepta cualquier tráfico (apagarla) o reiniciar un interfaz (encenderla).

**ping**

Se utiliza este comando para probar si un sistema en particular se puede contactar desde el sistema local, ping utiliza el ICMP para intentar alcanzar el sistema remoto, el cual no necesariamente debe estar en la misma red. Esto significa que usted puede utilizar ping para revisar que el enrutamiento entre las redes está trabajando correctamente y para revisar si un software TCP/IP está funcionando en el sistema remoto.

**netstat**

Se utiliza este comando para monitorear la actividades de red en el sistema local. Usted puede monitorear diversos aspectos de la red, incluyendo el estado de los puertos locales TCP y UDP, tablas de enrutamiento IP, memoria y el uso de otros recursos del sistema y estadísticas discriminada por protocolo del sistema local.

Otras utilidades están disponibles para proporcionar formas variadas de estadística referente al software de red. A medida que las redes llegan a ser más grandes y a estar distribuida sobre areas geográficas más amplia, se ha convertido más difícil utilizar estos métodos para manejar la red. Para poder administrar las redes complejas que son hoy común, se requiere un sistema más sofisticado de adminitración de la red.

***Adminitración Moderna de la Red***

Las herramientas modernas de la adminitración de la red han cambiado y se han adaptado con su misión así como escalabilidad. Utilizan utilidades de administración gráficas para controlar remotamente nodos de red. La admiistración global del sistema se ha convertido más genérica, es decir, no específico a un conjunto de sistemas o a un fabricante. Cada día se requiere menos y menos conocimiento de los detalles del productos de un fabricante pero si se requiere más conocimiento en los aspectos comunes entre los sistemas. Las herramientas basadas en SNMP aprovechan estas concordancias y las realzan.

Los requisitos de una administración moderna de una red incluye los isguientes puntos:

- Administración Remota.
- Disparar alarma cuando los componentes fallan o cuando ocurren acontecimientos anormales.
- Interfaz de administración común.
- Herramientas de administración integradas.

De éstos y de otros requisitos ha surgido un sistema de protocolos de administración de red que le permiten a administradores de red el control y el monitoreo del comportamiento de la red desde una consola de

administración. El protocolo de mayor intrés es el SNMP ya que este es parte del suite de protocolos del Internet.

***Componentes del SNMP***

Las herramientas de administración de red basadas en SNMP tienen muchas cosas en común, incluyendo:

- Utilidad de Administración.
- Software en el nodo para hablar con el manejador (agente)
- Descripción de las características del artículo que se manejará (MIB)

Estos componentes son apoyados por tres especificaciones:

- SMI (Structure of Management Information), que describe cómo identificar y describir objetos
- MIB (El Management Information Base), cual es la información que describe los objetos, por lo general almacenada en el nodo en cuestión.
- SNMP (Simple Network Management Protocol), que define la comunicación entre los administradores y los agentes.

Entre los componentes de una herramienta basada en SNMP, se incluyen:

| | |
|---|---|
| **Managed Objects** | Un objeto manejado es algún componente de la red. Es un concepto general pues un objeto manejado puede ser un sistema completo, una parte de un sistema tal como un módulo de enrutamiento o un NIC. Objetos Manejados pueden ser aún más abstracto, por ejemplo una máquina de venta por moneda; de hecho, cualquier cosa que se puede agregar a una red puede convertirse en un objeto manejado. |
| **Manager** | El Encargado es el software que sirve de interfaz al encargado de red humano. |
| **Agent** | Un agente es una pieza de software que puede comunicarse con un objeto manejado. Responde a las peticiones del encargado para retornarle la información sobre un objeto manejado o para alterar cierto atributo del objeto manejado. Este puede también transmitir la información al encargado sin ser preguntado, por ejemplo, para indicar una condición excepcional con respecto al objeto manejado. |

Para que estos componentes trabajen unidos, el sistema de administración de la red debe definir:

- Un medio de identificar los objetos manejados en la red.
- Un medio de especificar los objetos reales y sus atributos.
- Un medio de comunicarse entre el encargado y los agentes que controla los objetos manejados.

**Encargado, Agentes y Nodos.**

En la siguiente ilustración mostramos una posible distribución para ejecutar un sistema de administración de la red tal como la descrita. Cada uno de los sistemas en la red es un nodo manejado, con uno reservado como la estación de administración de la red. El software encargado (administrador) de red se ejecuta en este sistema.

**IMAGEN PAG 378**

Los nodos que son manejados no necesitan estar en la misma red que la del encargado; de hecho, el enrutador o el bridge que se utiliza para cruzarse entre las redes puede también ser un objeto manejado.

El software encargado de red puede ser basada en texto o CLI pero, pero hoy en día más comúnmente, se basa en entornos gráficos de alta calidad profesional.

**SMI (Structure of Management Information)**

Las especificaciones de nombre del objeto SMI se establecen en los RFCs 1155 y 1212. Los objetos son nombrados usando una estructura jerárquica. La puesta en práctica del SMI es el Management

Information Base (MIB).

Los MIBs de alto nivel son administrado por una autoridad central y la administración es delega en niveles más bajos. Los objetos en cada nivel son asignados números enteros. La identificación completa del objeto es una secuencia de números enteros. A los objetos también se le puede asignar nombres del tipo cadena, lo cual facilita la interacción con humanos.

Los objetos se nombran según el RFC 1155, “Structure of Management Information”. Esto define un esquema jerárquico para nombrar objetos, parecido al del esquema jerárquico usado en el DNS.

En cada nivel de la jerarquía, los objetos se asignan números enteros. Para especificar exactamente un objeto, usted concatena los números de los objetos que especifican una ruta con la jerarquía desde la raíz de la estructura al objeto sí mismo.

Esto puede conducir a nombres ciertamente complejos. Por ejemplo: ***1.3.6.1.2.1.5*** el cual es nombre del objeto que contiene la información sobre el ICMP.

Por esta razón, a los objetos también se les asignan nombres valorados del tipo cadena para que sean más fáciles de entender y recordar para los humanos.

La administración del espacio de nombre jerarquico puede ser dividido entre las autoridades centrales para los niveles más altos y las autoridades locales para los niveles más bajos. Esto es otra vez similar a la manera que se organiza el espacio de nombre del DNS.

***Espacio de Nombre del Objeto MIB (Management Information Base)***

El espacio de nombre jerárquico completo es grande. En esta ilustración que sigue se muestra la parte que conduce desde la raíz a la porción del MIB del Internet.

**FOTOOOOOOOOOOOOOOOOFOTOOOOOOOOO-380**

La raíz no tiene nombre, pero si tiene tres descendientes inmediatos que representan las organizaciones de control principal.

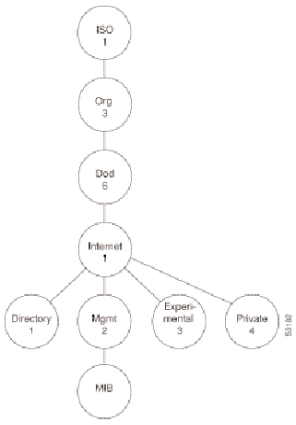
En cada nivel, los objetos individuales se asignan números enteros. Los objetos internos en el árbol del espacio de nombre se utilizan solamente para los propósitos estructurales. Los (leaf nodes) nodos hoja contienen los atributos.

La información relacionada con Internet se puede encontrar bajo nodo que tiene el nombre de 1.3.6.1. También puede ser referida usando el nombre Internet. Los subobjectos pueden ser nombrados relativoa al objeto Internet. Por ejemplo: ***Internet 2*** se refiere al subobjecto manejado (nombre completo 1.3.6.1.2).

***Internet MIB***

Management Information Base, es una base de datos donde se guarda toda la información relativa a la gestión de la red. El MIB tiene una estuctura en árbol, donde en la parte superior se encuentra la información más general sobre la red, y conforma avanzamos por las ramas se consigue información más específica y detallada. Cada nodo del árbol MIB se conoce como variable, y la parte superior del árbol MIB se denomina "Internet". Aunque la ISO ha definido un MIB modelo, cada fabricante de equipos tiene el suyo propio, e el que en general aunque las variables se denominen de diferente manera la información contenida en ellas es la misma.

El RFC 1155 define la estructura del MIB principal del Internet, el sistema de los objetos que se pueden interrogar con respecto a los protocolos de TCP/IP. Éstos están situados bajo objeto del MIB nombrado 1.3.6.1.2.1. El MIB define los objetos siguientes:

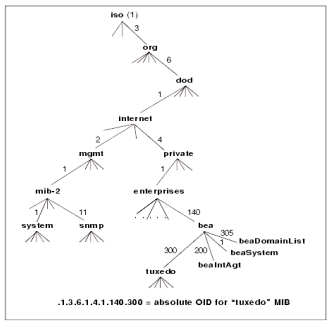



| | |
|---|---|
| System | Identificación e información sobre sistemas en la red |
| interfaces | Información sobre interfaces de la red |
| at | Conversión de dirección |
| ip | Información sobre IP |
| icmp | Información sobre ICMP |
| tcp | Información sobre TCP |
| udp | Información sobre UDP |
| egp | Información sobre EGP (protocolo dinámico de administración de la tabla de enrutamiento) |

El MIB del Internet es administrado por el Internet Architecture Board (IAB). Todos los manejadores de SNMP entienden el MIB del Internet; por lo tanto, pueden recopilar información de los nodos manejados en la red según este modelo.

La pieza system del MIB incluye objetos como son:
***system.sysdescr.0 1.3.6.1.2.1.1.1.0 system.syscontact.0 .3.6.1.2,1.1.4.0***
***system.sysobjectid.O 1.3.6.1.2.1.1.2.0 system.sysname.0 .3.6.1.2.1.1.5.0***
***system, sysupti inc. 0 1.3.6.1.2.1.1.3.0 system, sys! ocati on. 0 .3.6.I.Z.I.1.6.***

***Privado MIB***




Los MIBs privado son utilizado por los fabricantes para definir los objetos que representan sus propios productos. Bajo el objeto llamado privado (dirección 1.3.6.1.4), el manejo del MIB se delega a los fabricantes de hardware y software de red. Los fabricantes pueden definir objetos sus el propios basados en sus propios productos.

Cuando uno de estos productos se adquiere y este debe tener soporte de SNMP (o otra sistema de administración de red), entonces este incluirá un agente y una especificación del objecto(s) MIB que representa el producto. Esta información debe ser proveída de tal manera que pueda ser alimentada al software manejador de modo tal que el manejador pueda administrar el nuevo producto.

***SNMP***

El SNMP es un protocolo para la comunicación entre un manejador de red y los agentes que controlan objetos manejados. Está diseñado para ser simple, introduce el menor monto de sobre peso posible sobre la red. El manejador utiliza el puerto 161 y el agente utiliza el puerto 162. El SNMP es un protocolo simple, que usa UDP como su protocolo de transporte. Todos los mensajes SNMP están diseñados para ser transportados dentro un solo datagrama UDP.

Un manejador puede solicitar información de un agente acerca de un objeto manejado o puede establecer ciertos atributos de los objetos manejados. El agente responde a estas peticiones con la información o los mensajes de diagnóstico. El agente puede también enviar mensajes no solicitados (llamados traps) para indicar condiciones excepcionales en el objeto manejado.

***Mensajes del SNMP***

Cinco tipos de mensajes son definidos en SNMP que permiten a un administrador interrogar o establecer el valor de una variable, permitir a un agente contestar con el valor de una variable solicitada o permitir que un agente indique cierto evento relacionado a su objeto manejado. Estos son:

| | |
|---|---|
| **GetRequest** | Trae uno o más valores desde una variable específica de un objeto. |
| **GetNextRequest** | Trae el valor(es) sin especificar el nombre exacto; usando este puede iterar a través del MIB completo |
| **SetRequest** | Establece a variables específicas a valores especificados |
| **GetResponse** | Respuestas a un get o set request (simplemente llamados Response en SNMP 2 y el 3) |
| **Trap** | Indica cierto evento en el agente; enviado al manejador(es) especificado |

El mensaje GetNext permite a un manejador iterar por los valores de variables en un objeto sin saber el formato exacto del objeto. Este puede ser útil para interactuar con objetos privados que no esten completamente especificados en el MIB.

Las peticiones (Request) que especifican variables utilizan el esquema de nombramiento numérico jerárquico, descrito anteriormente, para identificar las variables y los objetos. La definición de la sintaxis de los objetos describe el formato de los datos que están siendo transferidos o con el GetResponse o el SetRequest.

Los “traps” pueden ser eventos, tales como: un sistema que se inicia en frío, un interfaz de la red fallando o un problema de enrutamiento.

***Comunidades***

Los agentes SNMP se pueden agrupar en comunidades. Para hacer una petición (para get o set un valor) de un agente, el manejador debe saber el nombre de la comunidad y especificarlo con la petición.

Esto introduce una forma primitiva de seguridad en el SNMP. En los años más recientes, se ha efectuado actualizaciones al SNMP que permiten mejoras y refinamientos en su funcionalidad. También hay actualizaciones de definiciones para los protocolos.

El nuevo SMI, definido en el RFC 1902, amplia los tipos de datos y mejora “get” y “set”. En esta manera, el SMI no fallará si uno de los valores no está disponible y el SMI puede adquirir un gran conjunto de valores a través de get-bulk.

La especificación también permite un formato mejorado de “traps” con el manejo de error mejorado.

También se provee para la administración de niveles múltiples. Esto establece una jerarquía entre los manejadores por la que un manejador general controla a los manejadores más bajos en al jerarquía usando “inform-request”. Los agentes Proxy pueden también controlar redes aisladas o las redes que no son SNMP.

Lo mejor de todo es que las nuevas especificaciones son compatibles con la versión 1 del SNMP.

El SNMP 2 fué concebido para mejorar las capacidades de SNMP 1 pero aún permaneciendo compatible con las versiones anteriores.

Esta define nuevos tipos de datos, mejora “get” y establece los mensajes, proporciona un mejor manejo de errores para las “traps” y define niveles del control de la administración de los sistemas del SNMP.

Desafortunadamente, la única área en la cual el SNMP 2 no fué éxitoso consiste en completamente definir una especificación apropiada de seguridad para substituir el uso de las cadenas de la comunidad como mecanismo de la seguridad.

## Versión 3 del SNMP

En esta sección destacamos algunas de las características de la versión 3 del SNMP, establecido en los RFCs 2271, 2272, 2273, 2274, y 2275. El SNMP 3 es la última versión de SNMP. Su objetivo principal es proporcionarle SNMP 2 y seguridad avanzada.

Con este fin, le agrega al SNMP 2, integridad de los datos, autentificación, cifrado (encriptación) y tecnologías claves de administración, todo esto junto con algunas mejoras administrativas.

Es muy probable que el SNMP 3 sea la última versión del SNMP por un largo tiempo. Esta disposición fué lanzada en el 1998 y ya existen muchos dispositivos que soportan SNMP 3.

La versión 3 agrega seguridad avanzada y cifrado atraves de la integridad de los datos, autentificación, ataques de enmascaramiento, así como MD5 y Secure Hash Algorithm. Agrega privacidad a traves de del soporte para DES enriptación de llaves privadas, autorización y el control de acceso.

Además, la versión 3 soporta el cifrado de llaves públicas, administración de llaves de seguridad y la interoperabilidad de múltiples versiones. Es compatible con las versiones 1 y 2.

### *Usar el SNMP*

Hay básicamente dos tipos de utilitarios SNMP: herramientas que se ejecutan en la línea de comandos (e.j., snmpget y snmpset) y los productos con GUIs.

Los productos gráficos son casi siempre de muy alta calidad. Estas gráficas por lo geeneral se ejecutan en estaciones de trabajo. Estas herramientas con GUIs ofrecen varias características que las hacen muy útiles, ofrecen fácil manejo y aprendizaje y hacen un magnifico interface a las herramientas SNMP. Sin embargo, estos productos GUIs privados tienden a ser costosos. Algunos ejemplos incluyen NetManager del sun, a NetView/6000 y Tivoli de la IBM, y HP OpenView.

## Otros Protocolos De la Dirección De la Red

El SNMP, aunque el estándar más robusto y extensamente el más usado para la solución de problemas y la representación de la red, este no es de ninguna manera la única herramienta de administración de la red disponible. Otros incluyen CMIP, CMOP y NetView.

La ISO está desarrollando un protocolo de administración de red conocido como CMIP (Common Management Information Protocolo) que es muy similar al SNMP pero que trabaja con la pila de sieta capa de la ISO. CMIP también está disponible en las redes de TCP/IP, conocido como CMOT; sin embargo, esto ahora ha sido adoptado y sobre pasado por eventos y substituida por SNMP.

Hace ya muchos años que la IBM ha tenido una estrategia de la administración de la red como parte del SNA, ésta era conocida como NetView (no debe ser confundida con los productos NetView/6000 que so basados en el SNMP). A medida que el SNMP es más aceptado, los productos NetView ahora son más integrado con el SNMP.

Observe que las definiciones de SMI y de MIB no son tales que están atadas exclusivamente al SNMP. Si se utilizan metódos de administración de redes alternativos, las específicaciones de los mecanismo de los objetos pueden ser utilizadas donquiera.

**Ejercicio 5-2: Implementación del Nuevo Producto de Administración de Red.**

No se proveen soluciones para este ejercicio.

Su tarea como administrador es planificar la implementación del nuevo producto en toda la sede principal de su empresa. Le será necesario tomar en consideración los siguientes puntos:

1.- ¿Cuáles parámetros del sistema y de la red dene usted medir?

2.- Cuáles son los nodos críticos en la red existente en la compañía en este momento?

3.- ¿Dónde tiene sentido colocar la estación de trabajo que efectuará la administración?

4.- ¿Existen algunas situaciones de red comunes que usted puede programar cierto tipo de respuesta automatizada?

## RESUMEN

En este capítulo, examinamos los siguiente:

- Solución de Problemas
  - Métodos metódicos de la Solución de Problemas
  - Los archivos protocol(s), services y inetd.conf
  - Herramientas y métodos útiles para la solución de problemas de la red que incluyen ping, traceroute, netstat, ifconfig, arp, nslookup y hostname.
  - El ICMP y los tipos de mensaje del ICMP

Administración de la Red

- Varios métodos y razones para la administración de la red
- Componentes y estándares de SNMP
- La necesidad del SNMP y de otras herramientas de la administración de la red

**PREGUNTAS**

Las respuestas a estas preguntas está en el Apéndice A.

1.- ¿Cómo puede usted comprobar si cualquier parte del TCP/IP ha registrado un error?

2.- ¿Cómo puede usted probar su espacio de nombre (namespace) de dominio?

3.- ¿Cuáles son algunas herramientas para la administración de una la red moderna?

# CAPITULO 6

*REDES DEL SISTEMA OPERATIVO GNU/LINUX*

# DNS (SISTEMA DE NOMBRE DE DOMINIO)


| TÓPICOS PRINCIPALES | No. |
|---|---|
| Objetivos | 213 |
| Preguntas Pre-Examen | 213 |
| Introducción | 215 |
| DNS a Simple Vista | 220 |
| Archivos y Comandos de Resolución de Nombres | 221 |
| Resumen | 230 |
| Pregunta Post-Examen | 241 |


## OBJECTIVOS

Al finalizar este capítulo, usted podrá:

- Instalar y configurar los servicios básico de DNS.
- Entender y ser capaz de implementar un Servidor de DNS.
- Detallar las razones dettras de la implementación de la identificación de un DNS, el uso de DNS para la identificación en los procesos de busquedas y los pasos tomados en el proceso de resolución de nombres así como la búsqueda inversa.
- Definir los elementos de un DNS y sus interacciones.

## PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. De dos razones del porqué el servicio de nombre no es efectuado en una forma centralizada.

2. ¿Cúal es el nombre del directorio raíz del árbol del DNS?

3. ¿Qué es un FQDN (Fully Qualified Domain Name)?

4. ¿Qué es un resolver?

5. ¿Son las siguientes sentencias falsas o verdaderas?
   - El DNS es independiente de la topología de la red.
   - Un resolver siempre sabe la localización del servidor de nombre ráiz/root.
   - El resolver efectúa un cache de la data.

# INTRODUCCION

En este capítulo cubriremos las teorías del servicio de nombres y responderemos las inquietudes de porque usar un servicio de nombre. En principio cubrimo lo que es el DNS (Domain Name System), el cual es usado en las mayorías de los sitios web conectados al Internet; pero si consideraremos otros servicios de nombres de Internet usados muy escazamente como es el WINS. En este capítulo también describiremos lo que son dominios, namespaces y zonas de los dominios como también explicaremos como trabaja un DNS. También se discutira los tipos de servidores, record de recursos y la resolución de un nombre.

## DNS A SIMPLE VISTA

El DNS es el mecanismo por el cual las direcciones IP son mapeadas a nombres de los ordenadores. Los siguientes tópicos son discutidos en esta sección:

- Servio de Nombre
- DNS
- Records de Recursoso
- Resolvers/Resolución
- Cache para mejorar Rendimiento
- Mapeado de Direcciones a Nombres
- Relación DHCP y DNS
- Actualización Dinámica
- Configurar y Ejecutar un DNS
- Otros Servicios de Nombres

## Servicio de Nombre

Un servicio de Nombre es un servicio de información que mapea o convierte nombres a direcciones IP. El servicio es accesado transparentemente; al comunicarse con un host remoto, el usuario no se percata del procedimiento usado por el servicio de nombre para encontrar la dirección IP.

Un servicio de nombre traduce un simple nombre a una dirección IP.

**codigolibre.org ========>>>>> 206.42.200.137**

No es suficiente poderse comunicar usando direcciones IP unicamente. Aunque la comunicación con el uso de IP es totalmente posible, las direcciones IP son dificiles de recordar para las gente y tienden a cambiar, por ejemplo, si un equipo es cambiado de una red a otra aunque dentro de la misma LAN. Otro problema con el uso de direcciones IP en el futuro es con el cambio del protocolo IP al ipv6.

Los nombres nos libran de las restricciones que colocan los protocolos, y aún más importante, son más fáciles de recordar que las direcciones IP numericas.

### *Los Problemas de un Espacio de Nombres Plano*

En los dias de un Internet con sólo unos cientos de hosts, los nombres de estos host eran administrados por una autoridad central. El administrador de sistemas contactaba el Centro de Información de Redes con siglas de NIC (Network Information Center) con el nombre del nuevo host. El NIC procedería a efectuar una búsqueda en su base de datos de nombres en el archivo HOSTS.TXT, para determinar si el nombre ya existe. Si el nombre es único y no existe entonces se le asigna ese nombre al nuevo host y se añade al archivo HOSTS.TXT seguido de la dirección IP.

El archivo HOSTS.TXT luego era propagado a todos los sitios de Internet usando el servicio de ftp. Los sitios se conectaban regularmente para proceder a descargar una copia fresca con los ultimos cambios.

En resumen, los problemas con este mecanismos son:

**• Existe una gran posibilidad de conflicto de nombres.**

Elejir un nombre único es difícil las mayorías de las veces. Los nombres son limitados y tenemos que ser muy creativos para poder seguir supliendo nombres únicos. Este problema se agraba a medida que se incrementa el número de hosts en el Internet.

**• La administración centralizada de los nombres se convierte en un problema.**

Este no es sólo un problema de la autoridad administradora sino de comunidad en general. Nadie desea tener que esperar para que una nueva elección de un nombre sea declarada única y luego tener que descargar un nuevo archivo hosts. El el caso de la NIC no existía tal administración de los nombres.

**• Los problemas de distribución se incementan dramáticamente.**

Con todo el mundo teniendo que descargar el archivo hosts por lo menos todos los dias, se producía una carga innecesaria sobre la red y el archivo hosts se convirtió intolerable. El mantenimiento de su consistencia se torno imposible. Cuando descargabas el archivo HOSTS.TXT, ya otro host había sido agregado.

## DNS

El DNS es una base de datos que mapea nombres a direcciones. Esta base de datos es distribuida en todo el Internet, independiente de la topología de la red. Aquí podemos ilustrar la traducción de un nombre de un ftp a una dirección IP.

**ftp.codigolibre.org ==========> 200.42.200.17**

La función principal del DNS es la distribución y su consecuente ventajas son:

- Los hosts no tienen que descargar la dirección correspondiente a cada host en el Internet; ellas solo deberan ubicar la dirección de la máquina con la cual desean comunicarse en el momento necesario.
- La distribución de la administración: si uste agrega un nuevo ordenador a su red local o si sólo le cambia su nombre, usted no tendrá que informarle a la agencia central. Simplemente tendrá que alterar la base de datos de us DNS.
- Conveniencia.- Nombres conocidos por el usuario son más fácil de recordar que sus respectivas direcciones IP.
- Consistencia.- Las direcciones IP pueden cambiar pero los nombres permanecen constantes.
- Simplicidad.- Usuarios necesitan aprender solo un nombre para encontrar recursos ya sea en Internet o en una Intranet.

### *Espacio de Nombre de Dominio del DNS*

Provee la estructura jerárquica de una Base de Datos distribuida. Cada dominio tiene un único nombre. Los hosts de Internet están organizados en un árbol único ya que la jerarquía es independiente de la topología de la red fisica. El root domain esta en lo alto y se representa por un punto. Bajo el root domain se encuentra el top level o first level, puede ser un tipo organizacional como .org, ó una localización geográfica como .do. El Second Level son registrados para organizaciones individuales tal como kernel.org. En el second level pueden contener muchos subdominios y algún dominio contiene hosts (host que viene a ser una computadora específica dentro del dominio). Un FQDN (Fully Qualified Domain Name) describe la relación exacta de un host a su dominio.

El término name space (o espacio de nombres) se refiere a todos los nombres posibles. Un namespace plano limita este conjunto, mientras que uno jerarquico permite un conjunto a escojer de nombres casi ilimitados. Al trabajar con los DNS concentrese en los namespace y no las redes subyacentes.

### Una Jerarquía de Dominios

La mejor manera de describir un dominio es como un sub-árbol de un namespace. Cada nodo en el sub-árbol es nombrado por el una etiqueta.

Existen dos convenciones usadas para elejir dominios, organizacional y geografica:

**Organizational:**

**com** Organizaciones Comerciales (como es http://www.ibm.com)
**edu** Organizaciones Educativass (http://www.see.edu)
**gov** Organizaciones Gubernamentales (http://www.nasa.gov)
**mil** Organizaciones militares (http://www.army.mil, http://www.navy.mil)
**net** Organizaciones de Redes (http://www.verizon.net)
**org** Organizaciones No-Comerciales (http://www.codigolibre.org)
**int** Organizaciones Internacionales (http://www.nato.int)

**Geographical:**

**do** Republica Dominicana
**mx** México

Se permite usar mayusculas y minusculas para registrar l;os dominios pero esto no significan nada ya que todas son tomadas como minusculas, lo que significa que se escribes dos nombres uno toda minusculas y otro con mayusculas y minusculas se trataran como el mismo nombre. La etiqueta debe empezar con una letra y puede terminar con una letra o un digito, y sus caracteres de interior deben ser todos letras y digitos o guiones. La etiqueta no debe ser más de 63 caracteres.

***Nombres de Dominio***

Un nombre de dominio esta compuestop de todas las etiquetas desde la raíz, escrito desde la derecha hacia la izquierda, separado por un punto (.). Un nombre que termina con la raíz (root) es denominado como un FQDN (Fully Qualified Domain Name). Una etiqueta debe ser único dentro de su dominio padre.

Los nombres de dominios pueden ser especificados de dos maneras:

- Los nombres absolutos son expresados en relación a la raíz, asi que terminan con la etiqueta de la raíz, un ejemplo es: ftp.linux.org.
- Los nombres relativos representan la última etiqueta de un nombre de dominio incompleto; ellos son relativos a un origen bien conocido (normalmente es la raíz del dominio) o a una lista de dominios usada como una lista de búsqueda.

Fijese como la raíz del dominio está a la derecha. Los nombres de dominio se escriben de atras para alante, desde el más específico (el nodo más bajo) hacia el menos específico (la raíz).

***Servidor de Nombres***

Los servidores de nombres son los repositorios de información que hacen la base de datos. Cada nodo en el árbol tiene cierta data asociado con el. Un servidor de nombre (en inglés, nameserver) almacena la información para el espacio de nombre, el namespace. Se dice que es autoritario para la parte del namespace de la cual el administra y contiene toda su información.

La tarea esencial de un servidor de nombre es responder las peticiones basado en la información que el contiene.

***Delegación***

Cuando la administración de dominio.org ha sido delegada a Compañía X,entonces dominio.com puede delegar la administración aún más.

Al delegar administración, estamos realmente delegando lo siguiente:

- El poder de decisión sobre la elección de nombres; la elección de los nombres de subdominios no esta sujeta a elegirlos únicos globalmente; los nombres deben ser únicos sólo dentro del subdominio al cual pertenecen.
- El almacenado y mantenimiento de los nombres dentro de esa parte del espacio de nombre del dominio (namespace).

***Zonas/Zones***

La información accesada via el espaciode nombre de dominio es rganizada en unidades llamadas zonas/zones. Una zona contiene toda la información acerca del dominio excepto la información delegada a otros servidores de nombres. Un servidor de nombres puede ser la autoridad de muchas zonas. Aún más, estas zonas no necesariamente deben encontrar adjecentes dentro del espacio de nombre de dominio.

El servidor raíz, por ejemplo, sirve subdominios como son .org asi como sirve el dominio raíz (pero, si observan los dominios de los países como el .do); los mismos principios son usados para mapear direcciones a nombres.

El uso de los términos DNS se torna un poso confuso; las personas a menudo se refieren a las zonas como dominios.

Es requerido que los dominios accesibles desde el Internet sean soportados por lo menos por dos servidores de nombres. Estos es sólo por razones de confiabilidad. A menudo, los administradores de sistemas han obviado esta regla y han sido blanco de castigo por la comunidad que rigen el Internet.

***Servidores Primarios y Secundarios***

La información de una zona es replicada atraves de por lo menos dos servidores para garantizar confiabilidad. La data de la zona debe estar disponible aún cuando un servidor fracaza o durante un ruptura en la red. Los servidores primarias son a veces referidos como los servidores maestro/master.

Toda la data origina en los archivos maestro o los archivos de data que se encuentran repartidos en todos los hosts que usan el sistema de dominio. Estos archivos maestros son actualizados por los administradores del sistema local.

Los cambios de una zona son efectuados en el servidor primario. El servidor carga la copia maestra de su data desde los archivos maestro cuando este se inicia.

Un servidor secundario mantiene una copia de la data de la zona. Este revisa periódicamente el servidor primario para asegurarse que sus zonas están actualizadas. Si no lo están, este adquiere una nueva copia desde el servidor primario de las zonas actualizadas. En realidad los servidores no son de un tipo o de otro; un servidor puede ser el master de una zona y el secundario de otra.

Un cliente que peticiona información desde un servidor de nombres no esta conciente si ese servidor de nombres es un primario i secundario en esa zona.

***Otros Tipos de Servidores***

Hay tres tipos de servidores: solo capturadores/caching, reenviadores/forwarders y esclavos/slaves. Todos los servidores hacen cache de data, pero un servidor caching-only no es quien autoriza para esa zona. Este toma la carga de otros servidores atraves de la construcción de un cache para poder responder las peticiones. Un reenviador/forwarder maneja las peticiones en nombre de un servidor de nombre cliente. No se necesita ninguna configuración especial; todos los servidores son forwarder por naturaleza ya que otro

servidor les envia peticiones recursivamente. Esto es útil cuando las conexiones a redes externas son lentas o quizas costosas. Los servidores esclavos son muy parecidos a los forwarder excepto que no sale fuera de la red local si no hay una respuesta desde el forwarder. Esto es muy útil cuando no hay conexión a la red externa o cuando la máquina pasarela/gateway esta actuando como un firewall.

**Records de Recursos**

Toda la información almacenada o transmitida dentro de un DNS se encuentra en un formato estandar llamado Resource Record (RK). El RK fundamentalmente esta dividido en cuatro parte:

| | |
|---|---|
| **Domain** | El nombre de un dominio al cual el record se refiere |
| **Class** | La clase del record (por lo general es IN para el Internet) |
| **Type** | El tipo de record, e.j., para que es la información |
| **Information** | La data para este record |

En este ejemplo ilustramos el significado de cada parte de un record de recurso o RK.

| | | | | | |
|---|---|---|---|---|---|
| **200.42.212.137** | **-ADDR.ARPA.** | **IN** | **PTR** | **SRI** | **-NIC.ARPA.** |
| **domain** | | **class** | **type** | | **information** |

***Tipos de Records de Recursos***

Los DNS definen varios tipos de records de recursos. Entre los más comunes se incluyen:

- **A (dirección IPv4)**
  Indica la dirección que corresponde a un host en particular
- **AAAA (dirección IPv6)**
  Indica la dirección que corresponde a un host en particular
- **NS (Nameservcr)**
  Indica un servidor de nombre responsable de un dominio en particular
- **SOA (Start of Authority/Inicio de la Autoridad)**
  Designa el inicio de una zona
- **PTR (Puntero/Pointer)**
  - Permite nombres especiales a que apunten a otras localidades en el dominio
  - Usado principalmente para el mapeado de direcciones a nombres

Esta lista en ninguna forma es completa y existe muchos más tipos de records, aqui sólo mencionamos unos cuantos para ilustrar.

Primero se necesitan records de Direcciones/Address, los cuales mapean nombres a direcciones- la verdadera función de un DNS. De segundo, necesitamos records de Servidor de Nombres, con los cuales un servidor de nombres para el dominio en cuestión. Tercero, necesitamos un record de Inicio de Autoridad (Start of Authority) el cual designa el comienzo de una zona; este también contiene la información relacionada con la propagación de la información de la zona entre los servidores primarios y secundarios. Y para finalizar, los records de Punteros/Pointer serán usados para mapear las direcciones a nombres.

**Resolvers**

El resolver es como el cliente procesa entrada en la base de datos del DNS. Este extrae la información desde el servidor de nombres en respuesta a las peticiones del cliente. Para lograr esto, este debe saber como llegar a por lo menos un servidor de nombres.

El resolver normalmente es implementado como un conjunto de librerias vinculadas con aplicaciones que necesitan comunicarse usando nombres de equipos. Una aplicación simplemente llama una rutina que resuelve/resolver y espera que esta retorne: esta no se alerta de la existencia de la peticiones del servidor nombres que el resolver esta iniciando para su beneficio.

***Resolución de un Nombre***

Un proceso cliente desea comunicarse con otro proceso en ftp.descarga.abiertos.org. Para poder efectuarlo, se requiere la dirección IP de esa máquina en particular, y no solamente el nombre. Si se usa las API de sockets, nuestro proceso cliente llama la función gethostbyname(), la cual es parte de la libreria resolver.

1. La función resolver envia una petición a servidor de nombre local (el resolver determina a cual servidor de nombres usar buscando en su archivo de configuración local). La petición específica el dominio de interes (ftp.descarga.abiertos.org) y la información requerida (un record del tipo Address).

2. El servidor de nombre local primero revisa su propia información para ver si poseé la respuesta a la petición. El caso negativo, este envia una petición por el dominio ftp.descarga.abiertos.org al servidor raíz/root. En el caso que el servidor raíz tampoco tenga la información, si sabe de un servidor más cercano al domio de interes, asi pues responde con una referencia/referral a un servidor del dominio .org.

3. El servidor de nombre local envia una petición por el dominio ftp.descarga.abiertos.org al servidor de .org. Este servidor no tiene una respuesta y responde con una referencia/referral a un servidor en el dominio abiertos.org.

4. El servidor local envia una petición por el dominio ftp.descarga.abiertos.org por abiertos.org. Este servidor no tiene una respuesta y responde con una referencia a un servidor para el dominio descarga.abiertos.org.

5. El servidor local envia una petición por el dominio ftp.descarga.abiertos.org por abiertos.org. Este servidor si contiene la información requerida (lo que en este caso signfica que tiene un record del tipo Address para ftp.descarga.abiertos.org). El servidor retorna la dirección al servidor que hizo la petición.

6. El servidor de nombre local responde a la petción original del resolver.

El rutina resolver gethostbyname() retorna al proceso cliente la dirrección de ftp.descarga.abiertos.org.

**Uso de Cache para Mejorar Rendimiento**

Al igual que en un procesador o un disco duro, el uso de cache en un DNS mejora el rendimiento. El rendimiento mejora dramáticamente ya que el cache es revisado primero para responder una petición. El problema que se presenta con el cache, claro como con todo las tecnicas de cache, es que si la data original del servidor de nombre ha sido actualizada puede existir un retardo ya que la data en cache es una copia local. Este servidor responderá peticiones que puede estar desactualizada. Esta data es conocida como no autorizada. Los servidores primarios y secundarios son la fuente original de la zona y retornan data que si es autorizada. Un servidor secundario está autorizado para la zona, ya que este contiene la información total de esta zona.

Existe un mecanismo de timeout/vencimiento para eventualmente descarta data en cache. Asociado con cada record de recurso llamado TTL (Time to Live, tiempo de vida) que indica que tiempo le resta a esta data en cache antes de ser descartada.

Los diseñadores del DNS consideran que el acceso a la información es más crítico que actualizaciones instantáneas o garantía de consistencia. La mayoría de la data en el sistema cambiará lentamente, pero el sistema debe poder manejar los subconjuntos que cambian más rápido, en el orden de segundos y minutos.

**Mapear Direcciones IP a Nombres**

Algunas aplicaciones necesitan saber el nombre del ordenador al ser entregado la dirección IP. Este mapeado es efectuado usando el dominio especial, IN-ADDR.ARPA. Este dominio es configurado especialmente para mapear direcciones a nombres. Las direcciones en este dominio corresponden a direcciones IP pero aparecen alreves, porque esta es la manera en que los nombres de dominios de los DNS se escriben. Los nodos en este dominio son nombrados por los bytes de su dirección IP. Colocar imagen del árbol de nombre y su relación con IN-ADDR.ARPA.

Las aplicaciones y servicios que se ejecutan hoy requieren que un host que se conecte a ellos pase busquedas en reverso (reverse lookup), un término usado para describir el proceso de mapear direcciones IP a un nombre, para asi poder establecer una conexión. Los hosts que no pasan la búsqueda inversa pueden que tengan un error lijero de configuración o pueden ser mal configurados a propósito para enmascarar la identidad de alguien intentando violar nuestra seguridad o ejecutar un ataque DoS (Denial of Service).

**Relación DHCP y DNS**

Las siglas DHCP significan Dynamic Host Configuration Protocol, que en castellano es Protocolo de Configuración de Hosts Dinámica. El propósito del DHCP es permitir que un ordenador en una red IP extraigan su configuración desde un servidor o servidores, en especial servidores que no tienen información exacta del ordenador en particular hasta que la petición sea efectuada. El propósito general del DHCP es reducir el trabajo necesario para administrar una red IP grande. El servidor DHCP puede ser configurado entregar un número de IP al azar o un nombre de dominio a equipo cliente por un tiempo en específico, denominado a un alquiler. Una vez se apaga el cliente y el periodo de tiempo de alquiler expira, el servidor DHCP puede re-alquilar a otro cliente el mismo IP. Una excepción a esta regla es cuando un IP estaticco es asignado a un cliente en particular. Si el servidor DHCP asigna un nombre de dominio a un computador cliente, se necesitará un servidor DNS para resolver la dirección IP apropiada.

***Actualización Dinámica***

El término actualización dinámica es usado para describir el proceso de actualizar automáticamente los records. El RFC 2136 detalla el procedimiento dinámico de actualización. Si una zona está configurada para utilizar actualizaciones dinámica, es entonces importante no editar el archivo de zona manualmente. Si se presenta esta necesidad, es de suma importancia entonces, apagar el servidor de nombre y editar el archivo de zona y eliminar el archivo de diario (journal) de la zona antes de reiniciar el servidor de nombre.

Los pasos del maestro primario para autenticar la petición a quien peticiona son:

1. Un mensaje de petición actualizada es enviada desde un cleinte a un servidor local.
2. El mensaje actualizado será reenviado al Servidor Maestro Primario.
3. El Maestro Primario revisa los prerequisitos especificados en el mensaje.
4. El equipo que efectua la petición es validado (dependiente de la implementación hasta que las extensiones de seguridad son incluídas, probablemente en la dirección de quien efectua la petición). Luego de actualizar su base de dato, el servidor la escribe a memoria no volátil (o resescribiendo el archivo maestro original o en un diario/log actualizado). El SOA serial es incrementado.
5. Si se implementa el RFC 1996, el servidor puede enviar mensajes “DNS NOTIFY” a los servidores esclavos.

***Configurar y Ejecutar un DNS***

Decidir ejecutar un DNS depende en un número de factores. Si usted está conectado al Internet global, deberá ejecutarlo. Aunque es obligatorio ejecutar el protocolo para usar el Internet, sin el DNS fuese imposible usar los recursos del Internet sin el. Si esta planificando conectarse al Internet en el futuro es recomendado registrar su direcciones de red y empezar a ejecutar el DNS.

Si usted no esta conectado al Internet, puede que aún asi decida ejecutar DNS si su red interna (internetwork) es lo suficientemente amplia, dividida sobre una área geográfica amplia o redes por separado, o si es administrada por diferente grupo de personas o divisiones lógicas.

Para ejecutar un DNS cuando usted no está conectado al Internet, deberá configurar su propio servidor de nombre raíz/root. Estos serán contactados como parte del proceso de resolución.

***Otros Servicios de Nombre***

Hay distintos métodos disponibles para la resolución de nombres parecidas al de los DNS. WINS es uno de estos métodos alternativos de resolución de nombre que esta basado en NT pero también disponible en GNU/Linux como característica del suite de Samba. El protocolo LDAP (Lightweight Directory Access Protocol) es para proveerr un servicio similar al servicio de directorio NDS (NetWare Directory Services) o el Active Directory. Aunque estos servicios de directorios son todos interoperativos con el estándar X.500 para los servicios de directorios, pero NetWare y WINS incluyen características de componentes propietarias, sin embargo existe una implementación completamente libre y Open Source de LDAP, de nombre OpenLDAP, disponible para GNU/Linux.

***WINS***

Puede que encuentre en redes ejecutando servidores sobre plataforma PC, el servicio WINS. Directamente del nombre se puede deducir la empresa de la cual se origina y sobre que clientes y servidores se ejecuta casi en exclusividad. Los servidores WINS, por lo general equipos NT, resuelven peticiones de direcciones IP en la misma forma que los servidores DNS. Fué creado para un uso primordialmente dentro de redes LANs y con el crecimiento de redes distribuidas hoy dia es muy poco usado.

WINS tiene muchas similitudes con el DNS en la menera que este comparte información de manera regular y en la menra que la información caduca si no es refrescada. Fué diseñado para ambiente dinámicos y en este respecto comparte sus características con el DHCP (Dynamic Host Configuration Protocol).

Un servidor WINS aceptará un registro dinámico de un cliente y procederá luego a propagar el nombre a los servidores locales, si el cliente no re-establece registro de su nombre por un período de tiempo establecido, será removido del espacio de nombre del dominio. Con el uso en combinación de DHCP y WINS, un cliente puede ser removido desde una red y reconectado a otrra sin la necesidad de ser reconfigurado manualmente.

***ACAP y LDAP***

La IETF (Internet Engineering Task Force) dirige el desarrollo de nuevos protocolos para proveer una mejor estructura organizativa del Internet y sus usuarios. Uno de estos protocolos es el ACAP (Application Configuration Access Protocol). Este le concede acceso a las aplicaciones como los clientes de correos a otros servicios como digamos los libros de directorios de direcciones.

En este momento, el Internet no poseé un servicio de requisición a directorio para que el ACAP acceda. El DNS proveé el servicio de una búsqueda por el nombre de un host y nada más. La especificación OSI X.500 es un servicio completo de servicio de directorio que incluye un modelo de información, espacio de nombre y una infraestructura de autentificación pero si se le agrega al Directorio Access Protocol es muy

pesada para su implementación global.

El LDAP (Lightweight Directory Access Protocol) simplifica muchas de las operaciones X.500 y puede ejecutarse sobre las redes existentes TCP/IP. Esto hace que el X.500 este disponible sobre una variedad más amplia y el LDAP puede hasta ser ejecutado nativamente con su propia base de datos. LDAP proveé ACAP con una estrucutura de directorio. Esta puede prover libros de direcciones, tanto para compañías privadas como libros públicos basaados en servidores como el de Netscape (servidor de directorio) y four11 con su página web en (http://www.four11.com). LDAP también puede integrar redes TCP/IP con servicios propietarios de directorio commo son el de Novel NDS y Banyan StreeTalk.

## CONFIGURAR UN SERVIDOR DNS

En esta sección nos encaminaremos por los pasos necesarios básicos de configurar los aspectos variados de un servidor DNS. Cubriremos los aspectos prácticos de la implementación de un servidor DNS. Discutiremos los parámetros pertinentes a su dominio. También cubriremos la escritura a los records de recursos, instalar y configurar el software del servidor y depurar el servidor de nombre.

Los siguientes tópicos serán discutidos en esta sección:

- Instalar y Configurar un Servidor de Nombre DNS
- El BIND (Berkeley Internet Name Daemon)
- El Archivo de Inicio del Servidor de Nombre- BIND 4
- El Archivo de Configuración del Servidor de Nombre -BIND 8
- Archivos de Data (Archivos Maestros)
- Configurar el Servidor de Nombre
- Información para Depurar

### Instalar y Configurar un Servidor de Nombre DNS

Existen tres componentes necesarios para configurar y ejecutar un servidor de nombre. Estos son el software del servidor de nombre, el archivo de arranque/boot y los archivos maestros (archivos de data). El software del servidor de nombre de GNU/Linux es llamada named. El archivo de boot puede que no sea necesario en todos los sistemas. El archivo maestro contiene toda la información del dominio, el mapeado inverso de las direcciones para las direcciones IP del dominio, el mapeado inverso de las direcciones para el loopback y el archivo cache.

El servidor de nombre DNS es implementado ejecutando el daemon de nombre named, también conocido como el BIND.

### El Daemon de Nombre de Internet de Berkeley (BIND)

Desarrollado en la Unviversidad de California en Berkeley, el BIND se ha convertido en el servidor de nombre más usado. El es proveído en muchos ambientes y es parte integral de las mayoría de versiones o distros de GNU/Linux.

El BIND se encuentra ya bajo el control de la ISC y es mantenido y desarrollado activamente. La ISC también tiene otros proyectos, como un servidor de DHCP este puede ser encontrado en la página web http://www.kc.org/. La versión 4 del BIND no está siendo desarrollada ni mantenida actualmente. La revisión actual de la versión del BIND 4.9.8, pero esta es sólo una parte de la serie de correción de fallas/bug-fix.

Los desarrolladores y mentenedores de BIND, la ISC, recomiendan en lo absoluto que los usuarios migren hacia en BIND 8.x, el cual tiene muchas mejoras y características sobre la serie antescedente 4.x, y mucho menos problemas de seguridad y ofrece DDNS (Dynamic DNS). La versión 8 de BIND, DDNS, fué lanzada en mayo 1997; esta implementa actualizaciones dinamicas, notificación de cambio de DNS y

ambientes dinámicos. Esto significa que los servidores maestros pueden ahora notificar a los esclavos de cambios ocurridos. Además, las extensiones de sseguridad de DNS usan firmas digitales para autentificación y aplicar estas actualizaciones de seguridad.

El soporte de IPv6 fué introducido por primera vez en la versión del BIND 4.9.4 y se continúa mejorando en el árbol de la versión 8. El árbol más reciente es el del BIND 9 que fué lanzado estable y se encuentra en la versión del BIND 9.1.0. El BIND es configurado usando un archivo de arranque/boot y archivos de data local. El archivo boot define:

- El directorio en el cual se ejecutará el servidor de nombre
- Las zonas en las cuales el servidor de nombres es primario o secundario y el archivo que continen la data del servidor de nombre.

Los archivos de data contienen los records de los recursos para todas las zonas que el servidor de nombres es autorizado.

***Archivo Boot del Servidor de Nombre- BIND 4***

Aquí le presentamos un ejemplo de un archivo boot para la zona abiertos.org del servidor primario de un BIND 4.x.x:

```
;
; Archivo /etc/named.boot para abiertos.org
;
directory               /var/named
;
;                       dominio          archivo          o host
;-------------------------------------------------------------------------------------------------------
cache                   •                root.cache
primary                 abiertos.org     db.abiertos.org
secondary   descarga.abiertos.org   200.42.200.47   descarga.backup
```

Hay dos versiones diferentes de BIND comúnmente usada al día de hoy, aunque ya esta disponible la versión 9, que son: la serie BIND 4.x.x, la cual ha está en uso por mucho tiempo y la más reciente producida por la ISC, la serie BIND 8.x.x. La serie 4.x.x ya no está en desarrollo y se le aconseja a los usuarios migrar a la serie BIND 8.x.x. Esta nueva versión es mucho más configurable y tiene muchas mejoras de seguridad y correcciones de falla. Si se encuentra en la necesidad de convertir un archivo named.boot al nuevo formato, puede usar el utilitario de nombre named-bootconf que se encuentra dentro del paquete de BIND 8.

Debido a que el formato del archivo de control del BIND cambia entre las versiones 4.x.x y 8.x.x, le mostramos ambas variantes aquí.

Este archivo de arranque indica que:

- El aservidor encontrará su archivo de configuración en el directorio /var/named .
- El servidor cargará con la data del dominio root/raíz, el cual se almacena en el archivo root.cache.
- Este es el servidor primario para la zona abiertos.org, y el recurso de record para esa zona deben ser cargados desde el archivo db.abiertos.org.
- Este servidor también es un servidor secundario para descarga.abiertos.org (el cual se presume tiene su propio servidor primario); la data de la zona es cargada atraves de una transferencia desde el host 200.42.200.47 al momento de iniciar el servidor, y se mantiene un backup en el archivo local descarga.backup.

***Archivo de Configuración del Servidor de Nombre- BIND 8***

Aquí le presentamos un ejemplo de un archivo de configuración para la zona abiertos.org del servidor primario de un BIND 8.x.x:

```
;
; Fichero de zona para abiertos.org
;
; Minimo indspensable para tener funcionando un dominio
;
@    IN          SOA    sn.abiertos.org.          hostmaster.linux.bogus.  (
                        199511301                      ; Número de serie, fecha de hoy+n de serie de hoy
                        28800                          ; Tasa de Refresco, en segundos
                        7200                           ; Tasa de Reintento, en segundos
                        3600000                              ; Caducidad para secundario, en segundos
                        86400 )                        ; Tiempo de Validez para Clientes, en segundos
                    NS     sn.abiertos.org.
                    NS     sn.backup.abiertos.org.
                    MX     10 mail.abiertos.org.       ; Intercambiador de Correo Primario
                    MX     20 mail.backup.abiertos.org. ; Intercambiador de Correo Secundario
localhost           A      127.0.0.1
ns                  A      127.0.0.2
mail                A      127.0.0.4
```

Deben de observarse dos cosas sobre los registros SOA. sn.abiertos.org debe ser una máquina actual con un registro A. No es legal tener un registro CNAME para la máquina mencionada en el registro SOA. Su nombre no necesita ser sn, podría ser cualquier nombre legal de máquina. A continuación, en postmaster.abiertos.org deberá aparecer algo como postmaster@abiertos.org; esto sería un alias de email, o una cuenta de correo, donde la(s) persona(s) que realizan el mantenimiento de DNS deberían leer con frecuencia el correo. Cualquier email respecto del dominio ser´a mandado a la dirección aquí indicada. El nombre no tiene por que ser postmaster, puede ser cualquier dirección email legal, pero la dirección email postmaster funcionará bien.

Hay un nuevo tipo de RR en este archivo, el MX, o Mail eXchanger. Este indica el sistema de correo a donde mandar el correo dirigido a alguien@abiertos.org, pudiendo ser tambíen mail.abiertos.org o mail.backup.abiertos.org. El número que precede a cada nombre de máquina es la prioridad del RR MX. El RR con el número más bajo (10) es aquel al que el correo sería enviado primero. Si este falla, puede ser mandado a otro con un número más alto, que será gestor secundario de correo, como mail.backup.org que tiene una prioridad 20 aquí.

Con el BIND 8, el archivo de configuración es por lo general llamado named.conf y no named.boot, que se nombró en la versión 4 de BIND. Fijese también que el caracter de comentario ha cambiado del anterior (;) a nuevo más estandar antoración de comentario (#).

Como podrá notar el sintaxis del archivo de confiuración principal del BIND 8 es muy diferente. La suma de muchas otras opciones posibles entre las llaves permiten una gran variedad de posibilidades de configuración que no era posible en el pasado y también se incluyen utilidades de logging, (diarios escritos) muy poderosas, y facilidades de control de acceso que se puede aplicar en base a por dominio en el BIND 8. Este archivo de configuración efectúa la misma función que las del archivo boot en el BIND 4.

***Archivos de Data (Archivos Maestros)***

Aquí le presentamos un ejemplo de un archivo de data para la zona abiertos.org del servidor primario:

```
; ejemplo de archivo de data del servidor de nombre para la zona abiertos.org y la subzone descarga.abiertos.org.
STTL        86400
abiertos.org.        IN    SOA   sede.abiertos.org.      backup.total.abiertos.org.    (
                                                         53              : Número de version
                                                         10800           : Refrecar después de 3  horas
                                                         3600            : Reintentar después de 1 hora
                                                         432000          : Expirar después de 1 Semana
                                                         10800   )       : TTL Negativo  de 3  horas
abiertos.org.                IN    NS                  sede.abiertos.org.
abiertos.org.                IN    NS                  gis.abiertos.org.
sede.abiertos.org.           IN    A                   200.42.210.43
gis.abiertos.org.            IN    A                   200.42.210.44

descarga.abiertos.org.       IN    NS                  parcial.sede.abiertos.org.
sede.descarga.abiertos.org.  IN    A                   200.42.210.45
```

El formato del archivo de data de la zona es casi identico entre las versiones de BIND 4 y 8, como son todos los aspectos de configuración del DNS.

Los recursos listados en este archivo de record indican lo siguiente:

- Los records listados no deben permanecer en cache por más de 86,400 segundos (24 horas) al menos que sea especifícado.
- El host sede.abiertos.org tiene autoridad sobre la zona de abiertos.org.
- Hay dos servidores de nombre para la zona de abiertos.org: sede.abiertos.org y gis.abiertos.org.
- Las direcciones de los servidores de nombre.
- El host sede.abiertos.org es un servidor de nombre para la zona descarga.abiertos.org (delegamos la administración de la zona).
- La dirección de ese servidor de nombre.

*El formato del archivo de zona del BIND requiere que los FQDNs, esos especificando el dominio completo hasta la raíz, deben ser escritos con un punto al final. En realidad se está especificando el root, ya que el nombre de root es "", el caracter null.*

En la resolución de nombre, el record NS refiere el requiriente a otro servidor de nombre. Ha sabienda que el nombre del servidor de nombre no es suficiente. Para comunicarse con el, necesitaremos la dirección IP del servidor de nombre. Esta información es proveída como un record de Address cuando especificamos un servidor de nombre que va ha ser consultado. El servidor de nombre lo retorna como parte de su respuesta al efectuar referencia a otro servidor de nombre. Estos records de dirección proveen el mecanismo que permiten que los DNS funcionen correctamente.

***Configurar un Servidor de Nombre***

Para poder configurar un servidor de nombre en un sistema GNU/Linux, deberá efectuar los siguinetes pasos:

1. Crear el archivo maestro que contine los records de los recursos perteniente a su dominio. Estos incluyen:
    - La información de nombre a dirección del mapeado de los hosts en su dominio
    - El mapeado de dirección a nombre para los hosts en su dominio
    - Archivo cache que contiene la información referente al servidor root
    - El mapeado inverso de dirección para su loopback local (127.0.0.1)
2. Configurar un resolutor para su host para que las aplicaciones en su sistema usen el daemon del servidor de nombre.
3. Configurar su archivo /etc/named.conf con la información acerca de la localización y autoridad de los

archivos maestros.

4. Iniciar el daemon del servidor de nombre, named, y verificar que no cometió ningún error.

***Información de Depurar***

El daemon named proveé información atraves de mensajes que envia al syslog. Su PID se almacena en /var/run/named.pid . Para encender la opción de depurar/debugging, inicie named con la opción -d y enviele una señal:

```
kill -USR1 'cat /var/run/named.pid'
```

Para incrementar aún más el nivel de depuración les enviamos otros USR1. Para apagar por completo la depuración, efectué el siguiente comando:

```
kill -USR2 'cat /var/run/named.pid'
```

Existen varias tecnicas para ver y diagnosticar porque ciertos aspectos de su servidor de nombre no este funcionando apropiadamente:

- Examinar los mensajes del log por lo general se pueden accesar con tail /var/log/messages.
- Volcar la base de datos del servidor enviandole a proceso in.named una señal INT. Este volcado se escribe a un archivo nombrado named_dump.db en el direcotorio raíz del servidor de nombre o al directorio /var/tmp/, dependiendo de la versión de BIND que este usando. Para volcar la base de dato actual y el cache al disco ejecute el siguiente comando:

  ```
  kill -INT 'cat. /var/run/named.pid'
  ```
- Ejecute in.named en modo de depuración/debug usadno la opción -d. La salida es escrita a un archivo de nombre named.run, como anteriormente o al direcotorio root del servidor de nombre o al directorio /var/tmp.
- Incrementar el nivel de depuración del servidor ya ejecutándose enviandole la señal USR1. Esta salida es escrita al archivo named.run. Mientras más elevamos el nivel de debug, más mensajes y la duración del mensaje se torna. Para detener el modo de debugging por completo, se le envia la señal USR2.

## Ejercicio 6-1: Configurar un Servidor DNS

Escriba los archivos de configuración del servidor de nombre en un subdirectorio de su home. Necesitará el la ruta completa a este directorio para colocarla luego en lugar de (nombre_de_Directorio). No se proveen soluciones para estos ejercicios.

1. Escriba el archivo de arranque/boot, named.conf, para su dominio:

```
# BIND v8 named.conf para el servidor primario de la zona Dominio
# opciones del servidor
options {
        directory "nombre_de_Directorio";
};
# zonas
zone "." in {
        type hint;
        file "root.cache"
};
zone "0.0.127.in-addr.arpa" in {
        type master;
        file "db.127.0.0";
};
zone "Dominio" in {
        type master;
        file "db.Dominio";
};
```

2. Escriba el significado de cada línea menos las secciones que son comentarios.

### Ejercicio 6-2: Archivo de cache del Servidor Raíz/Root

A los archivos de data le podemos dar cualquier nombre, siempre y cuando el archivo de configuración named.conf se refiere a el por ese nombre. Soluciones a este ejercicio se encuentran en el Apéndice B.

1. Cree el Servidor Raíz/Root Server, archivo será root.cache:

```
;   root.cache
;  pre-cargar el servidor primario con la información
;   a cerca de los sservidores root
                    99999999        IN      NSRootSrv.
RootSrv.            99999999        IN      A       RootIP
```

*Tome en cuenta que el RootSrv debe ser el FQDN del servidor root (como es ns.abiertos.gov.).*

2. ¿Por qué necesitamos un record A (Address) para el servidor root?

### Ejercicio 6-3: Archivo Data de los Hosts (Opcional)

No se proveen soluciones a este ejercicio.

Escriba el archivo data de los hosts para su subdominio, web.Dominio.org:

```
;   información del host para la zona Dominio.
;
$TTL        86400
Dominio.    IN    SQA SrvPrimario.     postmaster.SrvPrimario.
                              1;   Serial
                              10800;   Refrescar después de 3  horas
                              3600;    Reintento después de 1  hora
                              604300;   Expira después de 1 semana
                              86400;  TTL Minimo de 1 día
Dominio.                      IN  MS  SrvPrimario.
SrvPrimario                   IN  A   IPPrimario
;  información de los hosts individuales...
host1.Dominio.                IN A IP_host
```

Si se le dificulta entenderlo, hechemos un vistazo al siguiente ejemplo:

```
; información del host para la zona xyz.edu
$TTL                86400
xyz.edu.            IN  SOA     escuela.xyz.edu.        postmaster.xyz.edu.   (
                                                    1;          Serial
                                                    10800;   Refrescar después de 3 horas
                                                    3600:    Reintentar después de 1 hora
                                                    604800: Expira después de 1 semana
                                                    86400):  Minimo TTL de 1 día

xyz.edu.            IN      NS    escuela.xyz.edu.
escuela.xyy.edu.    IN      A           200.42.200.43
; Información de los hosts individuales
pop.xyz.edu.        IN      A           200.42.200.47
matematica.xyz.edu. IN      A           200.42.200.48
fisica.xyz.edu.     IN      A           200.42.200.49
quimica.xyz.edu.    IN      A           200.42.200.50
```

El record SOA (Start of Authority) indica que el host escuela.xyz.edu tiene autoridad sobre la zona xyz.edu. Es un record de recurso único, que se define sobre unas cuantas líneas usando parentesis. También

específica la dirección de correo del administrador responsable de la zona (postmaster@xyz.edu), con la @ reemplazada con un punto.

El record NS lista los servidores para la zona xyz.edu; estamos usando solamente un servidor. El record A es un record glue. Sin este record los servidores de nombre supieran el nombre del servidor autorizado, pero no su dirección, y por esto no pudiesen comunicarse con el.

Finalmente, hay records del tipo Address para cada host en la zona; esta es la información que estamos realmente interesados y es la razón principal de la existencia del DNS.

**Ejercicio 6-4: Archivo del Host Local (Opcional)**

No se proveen soluciones para este ejercicio.

1. Escriba el archivo del host local , backup.127.0.0.

   El archivo del host local mapea la dirección de la interface loopback local (la cual normalmente es 127.0.0.1) al nombre "localhost". Recuerde que el record SOA (incluyendo el postmaster) todos deben estar en una misma línea:

```
$TTL        86400
0.0.127.IN-ADDR.ARPA.      IN  SOA   SrvPrimario.   postmaster.SrvPrimario.    (
                           1;  Serial
                           10800;   Refresca después de 3 horas
                           3600;  Reintenta después de 1 hora
                           604800;  Expira después de 1 semana
                           86400;   Minimo TTL de 1 día
                           )

0.0.127.IN-ADDR.ARPA.      IN   NS     SrvPrimario.
SrvPrimario.               IN   A      IPPrimario

1.0.0.127.IN-ADDR.ARPA.    IN   PTR    localhost.
```

2. Iniciar el servidor de nombre:

```
# cd   /etc/rc.d/init.d
# named start
```

   Revise el archivo log (/var/log/messages) para ver si el servidor de nombre si inicio correctamente o si hubiesen errores en sus archivos de configuración.

## ARCHIVOS DE RESOLUCION DE NOMBRE Y COMANDOS

En esta sección, revisaremos paso a paso varios aspectos de la configuración y los comandos del cliente DNS. En esta sección cubriremos los aspectos practicos de la configuración del software cliente para tomar las ventajas de un servidor. En esta sección también cubriremos los comandos usados para interactuar con la base de datos del DNS para recibir la inforrmación y diagnosticar y reparar servidores DNS cuando presentan problemas.

Los siguientes tópicos son discutidos en esta sección:

- Resolución de Nombre
- /etc/hosts
- /etc/resolv.conf
- Orden de Consulta
- /etc/nsswitch.conf

- /etc/host.conf
- Herramientas y Resolución de Problema

***Resolución de Nombre***

Para un cliente poder conectarse a un host remoto, este debe convertir el nombre del host a una dirección IP. La resolución de nombre es la habilidad de un ordenador de interpretar un nombre de dominio como es www.dominio.org y determinar su dirección IP. La libreria resolutora en el cliente efectuá esta tarea. Existe un número de archivos de configuración que controlan la manera en que funciona la libreria resolutora.

| | |
|---|---|
| **/etc/hosts** | Este archivo contiene los hosts que no están listados en el DNS o que prefieren resolver sin el uso de un servidor de nombre. Este archivos es también para los clientes sin acceso a un servidor de nombre. |
| **/etc/resolv.conf** | Este contiene la lista de los servidores de nombres a consultar. Este puede contener más de uno; ellos serán consultados en orden. Si no hay ninguno presente, se consultará el localhost. |
| **/etc/host.conf** | Junto con archivo nsswitch.conf, este archivo determina cual método de resolución de nombre es usado primero usando el archivos hosts del cliente, un archivo hosts suplido por NIS o un servidor de nombre. |
| **/etc/nsswitch.conf** | Junto con host.conf, este archivo determina cual método de resolución de nombre es usado primero cuando se usa el archivo hosts del cliente, un archivo hosts suplido por NIS o un servidor de nombre. |

**/etc/hosts**

El archivo hosts provee la información pertinente a los hosts de Internet. El archivo hosts se encuentra en el directorio /etc en las distribuciones de GNU/Linux y es un archivo de texto plano que contiene las direcciones IP y los nombres de host de los clientes y servidores en la red interna y hasta en el Internet.

El contenido del archivo por lo general es parecido a la siguiente entrada:

```
127.0.0.1       miguel.midominio.org     miguel
```

El formato del archivo /etc/hosts es listar las direcciones IP primero, luego el número de nodo, seguido por el nombre alias. Se pueden incluir todos los alias necesarios. Si sólo un ordenador será agregado a este archivo, usted sólo necesitará la entrada de dirección loopback. Este es llamado el localhost.

El archivo hosts por lo general sólo contioene unas cuantas simple entradas, que incluyen el nombre y la dirección del host local. Esto es para permitir la dirección IP local que se encontrará cuando no se encuentra ejecutando un servidor de nombre, como por ejemplo cuando el ordenador se esta iniciando/booting (puede que ifconfig use el archivo hosts para determinar la dirección IP local).

También se puede suplir un archivo hosts via NIS (Network Information Service). NIS es un sistema de administración de redes que puede ser configurado para suplir archivos suplementarios a un sistema UNIX, incluyendo el archivo hosts. Usando NIS, un administrador de red puede distribuir un archivo de hosts único a múltiples clientes, asi proveiendo un servicio de resolución de nombre basado en la LAN sin el uso de un servidor de nombre.

El archivo hosts ya no es usado ampliamentre debido a que los servidores de nombre son muy comunes y son mucho más fácil de administrar que un archivos en cada cliente individual.

***/etc/resolv.conf***

Si el cliente no pudo resolver el nombre del host via el archivo hosts o no ha sido configurado para usar el archivo hosts, el próximo sistema lógico a resolverle su un nombre de host sería el servidor de nom-

bre. El archivos /etc/resolv.conf le dice a los rdenadores la localidad del servidor de nombre a consultar. El contenido del archivo /etc/resolv.conf por lo general luce como este que sigue:

Lo que sigue es un ejemplo del archivo resolutor para la zona de un cliente en la zona abiertos.org:

```
; resolv.conf para el cliente abiertos.org
domain          abiertos.org
search          abiertos.org
nameserver 200.42.200.42  ;  backup.abiertos.org
```

Hay tres items principales de información un resolutor necesita saber. El primero es el dominio por defecto- el dominio considerado local al host. El segundo es el dominio por defecto para agregarle a los nombres de hosts para que ellos asi no tener que especificar el FQDN para los hosts dentro de su dominio. La tercera entrada es la dirección del servidor de nombre del host local.

El archivo resolv.conf arriba mostrado indica que el dominio por defecto para este host es abiertos.org y el servidor de nombre que lo consulte debe ser enviado a la dirección 200.42.200.42 .

El archivo puede contener más entradas para que el servidor de nombre pueda consultar (por lo general alrededor de tres).

***Orden de Consulta***

El orden por el cual el resolutor consulta el DNS y el archivo local /etc/hosts o la desición si de hecho revisa el archivo local depende de la implementación de la configuración de la implementación del resolutor y de como los archivos de configuración locales están configurados. Los detalles de estas implementaciones no están definidas e la documentación de Internet de los RFC. El resolutor puede:

- Consultarprimero el DNS, luego revisar el archivo /etc/hosts si se retorna un error.
- No usar de ninguna manera el archivo /etc/hosts.
- Permitir que se efectue una elección via un archivo de configuración.

Este puede depender si se encuentra un NIS ejecutándose en el ordenador local. Por ejemplo, Sun Solaris 1.x no utilizaba el archivo /etc/hosts para nada y primero consultaba NIS.

Los sistemas GNU/Linux permiten que se efectue una elección via los archivos /etc/nsswitch.conf y /etc/host.conf.

***/etc/nsswitch.conf***

Un archivo de nombre /etc/nsswitch.conf determina el orden que las busquedas se efectuan cuando se requiere cierto tipo de información. El archivo nsswhch.conf es una base de datos del sistema. Este archivo también se encuentra en el directorio /etc. Se agrega una entrada en el archivo /etc/nsswitch.conf para cada base de datos. Por lo general estas entradas son muy simple, como por ejemplo “protocols: files” o “networks: files nisplus”. Pero, cuando son especificada múltiples fuentes, hay veces que es necesario definir precisamente las curcunstancias bajo las cuales cada archivo fuente será tratado. Los archivos deben ser ordenados con el servicio más usado en el principio.

Entradas válidas en el archivo incluyen:

| | |
|---|---|
| **nisplus o nis+** | Usar NIS+ (NIS versión 3) |
| **nis o yp** | Usar NIS (NIS versión 2), también conocido como YP |
| **dns** | Usar DNS |
| **files** | Usar los archivos locales |

| | |
|---|---|
| **db** | Usar los archivos de la base de datos local (.db) |
| **compat** | Usar NIS en modo compacto |
| **hesiod** | Usar Hesiod para la búsqueda de usuarios |
| **[NOTFOUND=return]** | Detener búsqueda si aún no se encuentra |

El contenido típico del archivo nsswitch.conf incluye:

| | |
|---|---|
| **passwd:** | files nisplus nis |
| **shadow:** | files nisplus nis |
| **group:** | files nisplus nis |

La primera columna es la base de datos y el resto de la línea específica como el proceso de búsqueda funciona. La especificación de la configuración para cada base de datos pueden que contenga dos cosas:

- Cuestiones específicas a servicios como “files” o “nis”
- La reacción del resultado de una búsqueda como “[NOTFOUND=return]”

En el caso de la resolución de nombre, una entrada puede ser asi:

**hosts: files nisplus nis dns**

La línea anterior primero consultaria el archivo hosts local, luego el servidor de NIS+, seguido por el servidor NIS y por último el DNS.

***/etc/host.conf***

El archivo host.conf contiene la información de configuración acerca de la libreria resolutor. El contenido debe tener sólo una palabra claver por línea con la información de configuración justamente al lado.

| | |
|---|---|
| **order** | Esta palabra clave específica como la búsqueda de host deben ser efectuadas. Este también tiene por lo menos un método de búsqueda, separado por comas si hay más de uno. Estos métodos son bind, hosts y nis. |
| **trim** | Esta puede estar listada multiple veces. Debe ser seguida por un único nombre de dominio, con un punto al final. Cuando establecida, la libreria resolv+ automáticamente recortará/trim el nombre de dominio dado del final de cualquier nombre de host resuelto via DNS. Esta es para ser usada con hosts locales y dominios. |
| **multi** | Los valores de esta palabra clave son on y off. Si establecida a on, la libreria resolutora retornará una dirección válida para que un host aparesca en el archivo /etc/hosts. El valor por defecto es off, ya que tiende ha ser más lento en sitios con archivos hosts amplios. |
| **nospoof** | Los valores de esta palabra reservada son on y off. Si esta opción se establece a on, la librería resolv+ tratará prevenir que el spoofing/fantasma de nombre de hos para mejorar el nivel de seguridad de rlogin y rsh. Luego de ejecutar una búsqueda de una dirección de un host, resolv+ efectuará una búsqueda de un nombre de host para esa dirección. Si dos nombres de host no se igualan, la consulta fracasará. |
| **alert** | Los valores de esta palabra clave son on y off. Si esta opción esta habilitada y la opción nospoof tambieen esta establecida, el resolv+ colocaría en el log una entrada de advertencia/warning del error via el utilitaro syslog. El valor por defecto es off. |
| **reorder** | Los valores de esta palabra clave son on y off. Si esta opción se establece en on, resolv+ intentará reordenar las direcciones host para que las direcciones locales esten listadas primeras cuando se ejecute un gethostbyname. El reordenamiento se efectua para todos los métodos de búsqueda. El valor por defecto es off. |

Un archivo /etc/host.conf típico luce asi:

```
order  hosts.bind
multi  on
```

La línea con la orden dice que primero busque en el archivo /etc/hosts antes de buscar el nombre del host usando un DNS. El BIND es el servidor DNS más usado y una parte de este es usada en los sistemas GNU/Linux para efectuar resolución de nombre en el lado del cliente. La palabra multi en la línea permite que líneas en el archivo /etc/hosts tengan más de una dirección IP. En otras palabras, este se mantiene buscando en el archivo hosts aún después de haber encontrado el mapeado correcto para ver si existen otros mapeos que también apliquen.

***Herramientas y Resolución de Problemas***

Ambos los administradores de sistemas y los usuarios a veces se encuentran tratando de diagnosticar y corregir un problema relacionado con la resolución de nombre, muchas veces sin percatarse de la relación. Hay muchas herramientas disponible para la corrección de problemas relacionados con resolución de nombre asi como para investigar información de domios, entre las cuales se incluyen:

- nslookup
- whois
- nslint
- dig

***nslookup***

El comando nslookup cuestionará al servidor de nombres especificado en el archivo /etc/resolv.conf acerca de la máquina especificada y devolvera su dirección IP. El formato de este comando es: nslookup máquina

En modo interactivo, nslookup puede hacer mucho más que sólo encontrar direcciones IP: puede preguntarle al servidor de nombres por cualquier clase de registros (no sólo A) e incluso puede mostrar la información referente a una zona entera.

Para entrar en el modo interactivo sólo es necesario teclear nslookup. El programa contestará con un signo de menor que ``>'' indicando que está listo para ejecutar comandos. Es posible entonces indicarle cualquier nombre de dominio y nslookup buscará por registros de tipo A. Para cambiar el tipo de registro que queremos encontrar es posible indicar set type=tipo, donde tipo puede ser cualquiera de los que ya mencionamos en secciones anteriores o incluso any, que indica cualquier tipo de registro.

La siguiente sesión, que se incluye como ejemplo, muestra cómo es posible encontrar información no sólo acerca de máquinas (direcciones IP), sino acerca de dominios (cuáles son sus servidores de nombres o de correo).

nslookup nos permite consultar interactivamente a un servidor de nombre. Distribuido con el BIND, debe estar disponble en todas las plataformas. Este comando se comunica con un servidor a la vez, ejecutando peticiones de un tipo particular de records de recursos desde un dominio especificado. Este no utiliza NIS o una base de datos de host.

Lo siguiente es un ejemplo del comando nslookup:

```
[root@proxy-ap root]# nslookup -sil abiertos.org
Server:         10.0.0.1
Address:        10.0.0.1#53
Non-authoritative answer:
```

**Name: abiertos.org**
**Address: 206.123.67.148**

Estos utilitarios son extensos y deben ser estudiados a fondo para poder dominarlos por completo.

***whois***

El comando whois es una herramienta útil para autentificar las cuentas de usuario y adquirir información sobre el servidor, incluyendo información de facturacion y un soporte telefonico. El comando alinia la sintaxis como sigue:

**whois nombre_servidor@whois_server**

El comando whois típicamente es usdo para identificar los usuarios, para locaizar nombres de dominios libre o hasta para revisar los records de un servidor DNS. Por ejemplo, si necesitamos saber el nombre de host de un usuario, el administrador puede revisar su servidor y si es necesario, llamar y verificar la cuenta. Un administrador también puede encontrar útil para verificar si un nombre de dominio que el desea utilizar aún esta disponible o no.

Existen unos cuantos servidores whois disponibles. Compañías, organizaciones y universidades todas deben registrar sus nombres de dominios con un servicio de registro de nombre (Domain Name Registration Service). Estas compañías registran y venden nombres de dominios. Ellos también mantienen una base de datos de todos los clientes que con ellos han registrado un dominio y los desplegan en su servidor whois. Como hay tantos servidores whois que solo contienen la información de los clientes de sus organizaciones, consultar el servidor para saber en cual servidor whois es que esta listado. Los dos servidores más amplios del Internet son whois.register.com y whois.networksolutions.com.

Red Hat utiliza una distribución de whois llamada fwhois. El fwhois no ofrece todas las opciones que el servidor whois provee. La gran mayoría de servidores whois permiten que palabras reservadas (como son Domain, Registrar, etc.) sean usadas para registringir la información retornada a campos especificos. De esta forma, es posible buscar un servidor por su nombre de dominio o buscar un servidor y sólo desplegar el nombre de dominio. Excisten un cliente para el X Window de nombre xwhois el cual es un cliente gráfico que despliega información en una ventana X Window y provee un interface de manejo intiutivo para usuarios sin experiencia con los servidores whois.

***nslint***

La herramienta nslint es útil para los administradores de servidores de nonbre ya que esta los asiste en identificar errores en los archivos de zonas. El uso del comando es realmente simple. Si su archivo named.conf se encuentra en /etc y las rutas están establecidas correctamente, sólo tendrá que ejecutar el comando sin ninguna opciones:

**# nslint**

Esta procederá a reportar cualquier error a pantalla. Si recibe un gran número de errores, puede que le sea más fácil direccionarlo por tubería a less o more para poder desplegarlo una página a la vez:

**# nslint |less o # nslint |more**

Si prefiere almacenar la informacióna un archivo log, podrá entonces direccionar la salida a un archivo:

**# nslint > ns_erorres.txt**

La herramienta nslint tiene varias opciones para afectar el compartamiento de su salida. Dos de las más usadas son:

-c Específica un archivo named.conf diferente al por defecto en el directorio /etc
-d Desplega más información de las cosas a revisar

Si su distribución no incluye la herramienta nslint, podrá descargarla desde el Internet.

***dig***

La herramienta dig es similar al nslookup. De hecho nslookup está deprecada y devemos usar dig como su reemplazo. La herramienta dig es usada para buscar información directamente desde los servidores de dominio. esta puede efectuar simple busquedas de DNS o ser usada para ayudar a diagnosticar y solucionar problemas de DNS.

La herramienta dig tiende a proveer más información que otras herramients de búsqueda de infomación de dominio. ejecutar dig con abiertos.org como argumento nos despliega una página de información:
$ dig abiertos.org

```
; <<>> DiG 9.2.2 <<>> abiertos.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12082
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;abiertos.org.                  IN      A

;; ANSWER SECTION:
abiertos.org.           4946    IN      A       206.123.67.148

;; AUTHORITY SECTION:
abiertos.org.           4639    IN      NS      ns1.quxnet.net.
abiertos.org.           4639    IN      NS      ns2.quxnet.net.

;; ADDITIONAL SECTION:
ns1.quxnet.net.         157352  IN      A       206.123.67.145
ns2.quxnet.net.         157352  IN      A       206.123.67.146

;; Query time: 512 msec
;; SERVER: 10.0.0.1#53(10.0.0.1)
;; WHEN: Wed Jun  8 14:56:05 2005
;; MSG SIZE  rcvd: 124
```

Si comparamos con la salida de nslookup de abiertos.org notamos que mucho menor:

```
$ nslookup  -sil abiertos.org
Server:         10.0.0.1
Address:        10.0.0.1#53

Non-authoritative answer:
Name:   abiertos.org
Address: 206.123.67.148
```

## RESUMEN

En este capítulo, cubrimos el Sistema de Nombre de Dominio (Domain Name System, DNS). Entre los puntos principales que tocamos se incluyen:

- Un servicio de nombre que traduce nombres de host a direcciones IP.
- El TCP/IP utiliza a DNS como su servicio de nombre.
- El DNS usa unsistema de dominio jerarquico donde cada dominio puede contener sub-dominios.
- La administración de los dominios de DNS es delegada a base de datos (llamadas zonas) que son distribuidas en todo el Internet.
- Los dominios pueden contener múltiples zonas.
- Cada servidor mantiene los records de recursos de su zona.
- Al configurar un servidor DNS, deberá configurar archivos maestros.
- Se recomienda utilizar herramientas administrativas para asistirle en escribir sus archivos maestros
- Existen otros servicios de nombre que son similares al DNS incluyedo el NIS de Sun entre otros.

## PREGUNTAS POST-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Son las siguientes acertaciones ciertas o verdaderas?

   • Cada host debe ejecutar un servidor de nombre individual

   • Un servidor de nombre puede ser primario a dos zonas.

   • Un host debe saber la localidad de un servidor de nombre de su zona padre

   • Un servidor de nombre debe servir por lo menos a una zona.

2. ¿Por qué es que alguna data es no autorizada (data nonauthoritative)?

3- Nombre un servicio de nombre alternativo.

4. Nombre tres tipos de records de recursos.

5. Hay dos convenciones usadas al elegir un dominio. ¿Cuáles son estas dos convenciones? Nombre dos ejemplos de cada una.

CAPITULO 7

*REDES DEL SISTEMA OPERATIVO GNU/LINUX*

# OTROS SERVICIOS DE RED


| TÓPICOS PRINCIPALES | No. |
|---|---|
| Objetivos | 238 |
| Preguntas Pre-Examen | 238 |
| Introducción | 239 |
| El inetd y xinetd | 240 |
| tcpd | 243 |
| DHCP (Dynamic Host Configuration Protocol) | 248 |
| Telnet | 250 |
| El SSH/OpenSSH | 253 |
| FTP | 254 |
| SQUID | 254 |
| NEWS | 254 |
| Sincronización del Tiempo | 254 |
| Protocolo RPC | 254 |
| Portmapper | 254 |
| NIS (Network Information Service) | 254 |
| LDAP | 254 |
| Resumen | 255 |
| Pregunta Post-Examen | 256 |


## OBJETIVOS

Al completar este capítulo, usted será cápaz de:

- Configurar un sistema GNU/Linux como un cliente o servidor DHCP.
- Configurar y usar un servidor anonimo FTP.
- Configurar un servidor telnet y explique poque es deseable deshabilitarla.
- Instalar un servidor Squid para habilitar el cache de páginas Web.
- Explicar como funciona un servidor de noticias (news).
- Habilitar la sincronización usando NTP.
- Configurar un cliente y un servidor NIS.
- Instalar y Configurar un servidor LDAP.

## PREGUNTA PRE-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1. Nombre tres clientes FTP.
2. ¿Qué significan las siglas LDAP y que hace este?
3. ¿Cuál servicio de red provee direcciones IP?
4. ¿Cuál es la diferencia entre NTP, ntpd, y xntpd?

# INTRODUCCION

Una de las fortalezas de GNU/Linux es su habilidad de proveer una variedad de servicios de red. En este capítulo, cubriremos varios servicios que tarde o temprano usted necesitará implementar, entre estos se incluyen DHCP, telnet, FTP, Squid, NIS, y LDAP.

Los primeros dos servicios que se introducirán son servicios que proveen control y cierto niveles de seguridad para otros servicios:

- inetd El daemon que administra los otros servicios
- tcpd Un daemon de seguridad que provee control del acceso rudimentario

## INETD y XINETD

Amenudo referido como el Super Servidor o el Super Daemon de Internet porque de las muchas funciones que este puede efectuar, el inetd es una parte importante del sistema porque este ayuda a conservar recursos, actua como una primera línea de defensa contra los usuarios no deseados y puede restringir acceso al root por usuarios permitibles. Solo se puede encontrar un inetd a la vez y por lo general se inicia cuando el sistema primero se inicia desde los scripts locales en el directorio de control /etc/rc.local.

El inetd puede efectuar servicios internos simple como son echo, discard, generador de caracteres (chargen), daytime y time. Daytime es el tiempo presentado en un formato entendible por humanos. Time es el tiempo en un formato presentado para las máquinas en la forma del número de segundos desde la medianoche de, Enero 1, 1900. El inetd provee ambos servidores basados en TCP y UDP para estos servicios asi como para todos los otros.

Una de las funciones principales del inetd es que este puede actuar como un vigilante para los sockets (zocalos) Internet. El inetd monitoreá un número de sockets diferentes esperando que un usuario externo efectué un contacto. Una vez una conexión es hecha, el inetd decide a cual servicio el socket corresponde y lanza (spawns) un programa que puede manejar la petición.

El inetd puede manejar tramas y servicios de tipos datagramas. Si la petición es de servicios de tramas, después que inetd ha pasado el socket al servidor, regresara y escuchara por nuevas peticiones de coneccion. Si las petición es para servicios de datagramas, después que inetd ha pasado el socket al servidor, este tiene que esperar, ignorando la actividad del socket de datagramas hasta que el servidor exista. La opción de esperar puede ser utilizada para tramas y/o servidores multihilo solo debe ser cambiado en el archivo de configuración inetd.conf.

Otra función principal de inetd es que este aligera la carga sobre el sistema, inetd es sólo un demonio, que puede invocar a otros demonios cuando es necesario en vez de usar a un número de diferentes demonios permanentes y aislados. Esto libera recursos, por consiguiente disminuye la carga sobre el sistema,

Hay algunas desventajas a la utilización inetd a diferencia de la utilización de demonios aislados. Correr inetd quiere decir que siempre que una petición de conexión entre, inetd debe realizar un número de procesos diferentes para manejarlo. El inetd debe bifurcar un nuevo proceso y luego cargar al nuevo demonio. Por consiguiente, el nuevo demonio debe cargar y analizar (dividir el código sobre las más pequeñas partes que pueden ser analizadas) su configuración. Todos esto combinado crea un proceso más largo que si usted usara a demonios independientes. Cuando se usan demonios independientes, ellos reciben una petición, hacen una copia de sí mismo y la copia maneja la petición.

A pesar de esta desventaja, usted probablemente querrá usar inetd. La ventaja de usar inetd sobre pasan sus pequeña desventaja. El uso de demonios independientes desperdician muchos recursos del sistema, recur-

sos que los servidores pequeños no disponen para malgastar, inetd es generalmente un programa bueno y es una parte importante del sistema Linux.

Hoy en día existen dos demonios populares de Internet usados por distribuciones GNU/Linux:

- inetd
- xinetd

***El inetd***

Por muchos años, inetd ha sido el demonio de Internet usado por distribuciones GNU/Linux. Este usa un solo archivo de configuración, /etc/inetd.conf. Dentro del archivo/etc/inetd.conf, los servicios controlados por inetd son catalogados, uno por línea. En cada línea, hay siete campos. Para cada servicio configurado, debe haber una entrada en los siete campos. Los campos son así:

**service_name socket_type protocol wait/nowait user server command line**

| | |
|---|---|
| **service name** | Este es traducido a números de puertos buscandolos en el archivo de servicios(/etc/services) para los servicios TCP y UDP o el demonio portmap para servicios RPC. |
| **socket type** | Este debe ser el tipo de socket que el servicio usaria. Esto va a ser stream, dgram, raw, rdm, o seqpacket. Los servicios basados en TCP siempre deberían usar el tipo stream, mientras los servicios basados en UDP deberían usar dgram. |
| **protocol** | Este debe ser un protocolo valido encontrado en /etc/protocols. TCPMUX debe usar TCP. Normalmente TCP o UDP es usado. |
| **Wait/nowait** | Este le dice al servicio si debe procesar peticiones múltiples juntsd o no. El datagrama (dgram) debe esperar, mientras que el stream o el socket multihilo deberan usar nowait. |
| **User** | Este debe contener el usuario que el servidor debera ejecutar. Esto es como un servidor puede usar inetd para limitar el acceso con el root. Esto es hecho simplemente cambiando nombre del campo de usuario por un usuario de clase inferior o un usuario con menos acceso a diferencia de root. Un ejemplo de esto sería si el nombre fuera cambiado por invitado (guest). |
| **server** | Este debe contener el nombre de la ruta del program a a ejecutar por inetd sobre la conexión. Si el servicio es interno, entonces este campo simplemente debería ser “interno”. |
| **command_line** | Esto contendrá una línea de comandos o argumentos para inetd, Este campo debería comenzar con el nombre corto del programa, argv [0]; sin embargo, probablemente es ocultado por el shell. Otra vez, si los servicios son internos, entonces el campo debería leer “interno”. |

Por ejemplo, es asi como los campos deben aparecer en el servico Ftp:

**ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -1 -a**

Siempre que el archivo de configuración de inetd sea cambiado, el demonio tiene que releer el archivo antes de que los cambios surtan efecto. Las opciones siguientes permitirán a inetd para releer el archivo:

| | |
|---|---|
| **-c** | Este fuerza inetd a releer el archivo de configuración. Este envía la señal de-SIGHUP al inetd actualmente corriendo. |
| **-k** | Este comando mata el inetd ejecutándose actualmente. Este envía el comando de -SIGTERM a inetd, haciéndolo retirarse. Usted entonces debe reiniciar inetd. |

Las opciones siguientes pueden ser usadas con inetd, si usted es el superusuario:

| | |
|---|---|
| **-d** | Este habilita depuración (debugging). |
| **-q** | Esto establece el tamaño de la coloa/queue de espera del socket a un especificado valor. Por defecto es 128. |
| **-R** | Esta opción específica el número máximo de veces que un servicio puede ser invocado en un minuto. Por defecto es 1,000 por minuto. |
| **-l** | Este habilita o desabilita el log de conexión para inetd. Esto puede ser práctico porque inetd escribe al |

log todas las conexiones acertada o fracasada. Si hay un registro de una negación repetida del mismo sistema remoto, hay posibilidades que esten tratando de rompers nuestro sistema. El comando -l envía el comando de SIGQUIT al inetd..

***El xinetd***

El otro demonio popular de Internet es xinetd. Hace ya un tiempo que xinetd a progresivamente a empezado a ganarse un espacio en las distribuciones de GNU/Linux. " Desde el lanzamiento de Red Hat Linux versión 7, este es el demonio por defecto de Internet en una de las distribuciones más reconocidas en el mercado de GNU/Linux.

El formato de archivo de configuración para xinetd es diferente de él tradicional /etc/ inetd. Confudido por inetd los usuarios de xinetd usa una configuración de dos etapas: el archivo de configuración para el servidor xinetd sí mismo y un directorio que contiene los archivos de configuración individuales para cada uno de los servicios bajo su control.

Un archivo de configuración de /etc/xinetd.conf puede lucir así:

```
# archivo de configuración simple para xinetd
# Algunos por defecto, y incluye /etc/xinetd.d/
defaults
{
instances               = 60
log type                = SYSLOG authpriv
log_on_success          = HOST PID
log_on_failure          = HOST RECORD
}
includedir /etc/xinetd.d
```

Un archivo de ejemplo de configuración puede ser este de un ftp:

```
# default:   on
# description; The wu-ftpd FTP server serves FTP connections.  It \
# uses normal, unencrypted usernames and passwords forauthentication,
service ftp
(
socket_type             = stream
wait                    = no
user                    = root
server                  = /usr/sbin/in.ftpd
server_args             = -1 -a
log_on_success          += DURATION USERID
log_on_failure          += USERID
nice                    = 10 1
```

Muchos de los parámetros listados son similares a aquellos en el archivo /etc/inetd.conf, como son socket_type, user, server, y otros. El campo commandline de /etc/inetd.conf está presente como el campo de server_args en un archivo de configuración de servicio de xinetd.

### Ejercicio 7-1: Configurar Servicios en xinetd

Con el lanzamiento de la versión 7 Red Hat GNU/Linux, xinetd fué incluído y presentado como un reemplazo para inetd. Mientras que la configuración de ambos presenta ciertas diferencias entre xinetd y inetd, ambos son lo relativamente intuitivos para que nos podamos facilmente adaptar entre el formato de uno y el otro. En este ejercicio, demostraremos como poner en práctica cambios a un par de servicios dentro de la nueva estructura.

La soluciones a este ejercicio se incluyen en su contenido.

1. Arranque su sistema en Linux.

2. Ingrese como root.

3. Verifique que el ftp se puede ejecutarse para el localhost. Una vez que usted ve que la conexión no falló, presione CONTROL+C para salir del shell y regresar al prompt.
   **# ftp localhost**
   **Connected to localhost.**
   **220 localhost FTP server (Version wu-2.6.1(l) Ued Aug 9 05:54:50 EOT 2000)**
   **ready.**
   **530 Please login with USER and PASS.**
   **Name (localhost:root): #**

4. Ahora para deshabilitar el sendee, muévase al directorio del árbol de configuración del xinetd:
   **cd /etc/xinetd.d/**

5. Aquí usted encontrará los archivos que son usados para controlar los servicios. Use un editor de texto para abrir el archivo de configuración de ftp:
   **# vi wu-ftpd**

6. Agregue las siguientes directivas de configuración:
   **disable = yes**

   Debe ser dentro de los corchetes. Un lugar bueno para ello sería después de la directiva.
   **r* 1 r*i**

7. Guarde y cierre el archivo.

8. Reiniciar el super servidor xinetd:
   **#/etc/init.d/xinetd restart**

9. Intentar conectarse por ftp al localhost otra vez. El servicio ahora debe estar cerrado.
   **# ftp localhost**
   **ftp: connect: Connection refused ftp> 10.**

   Salga del ftp escribiendo simplemente“bye” en el prompt del ftp.

**El TCPD**

.

El demonio tcpd es una manera simple de añadir cierto nivel de seguridad de acceso a aquellos servicios controlados por inetd. Esto es una envoltura o wrapper TCP, que intercepta peticiones y las envuelve desde su aplicación estandar a su propia. La función de tcpd es la de supervisar las conexiones al servidor hechas vía inetd y luego permitir o negar la conexión basada en la información contenida en los archivos hosts.allow y hosts.deny.

Para que tcpd trabaje, es necesario efectuarle cambios menor al archivo /etc/inetd.conf. Para cada servicio supervisado por tcpd, el listado de la ruta de las aplicaciones deberá ser cambiado a la ruta usado por tcpd. Por ejemplo, la entrada normal para el servicio de telnet en /etc/inetd.conf es:

Después del cambio, la entrada se convierte en la siguiente:
**telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd**

This tricks inetd into running tcpd when a service is requested.

El tcpd entonces escribirá la petición al log y revisará las reglas de acceso en los archivos /etc/hosts.allow y /etc/hosts.deny, en ese orden, basado en el ID del usuario remoto. Una vez y tcpd encuentra la entrada aplicable para el host remoto y el servicio que este ha hecho la petición, el tcpd usa esa información para permitir o denegar la petición y no revisa los archivos más allá. Si no existe una una entra para el host ID en uno de los archivos hosts.allow o hosts.deny, entonces la conexión será permitida por defecto.

Los archivos hosts.allow y host.deny contienen un listado de los servicios y los hosts en un formato de servicio:nombre_de_host:comando, donde comando es un campo opcional. Un ejemplo típico del archivo hosts.allow es parecido al siguiente:

**ALL: miguel ivelis dessi in.ftpd:**
**LOCAL EXCEPT jazzy in.telnetd:**
**192.168.2**

La primera línea de este archivo hosts.allow nos dice que los hosts remoto miguel, ivelis y dessi tienen aaceso a todos los servicios ejecutándose en el host local. Las máquinas que se encuentran en el mismo dominio no tienen que presentarse con sus nombres de host completo. El uso de ALL en el campo servicio indica que cada servicio en el host está disponible a esos usuarios remotos.

La entrada ftp es muy interesante. La palabra LOCAL indica que todas las máquinas en el mismo dominio, esas que cuales sus nombres de host no están separadas por un ".", tienen acceso al ftp. El uso de la palabra EXCEPT indica que la máquina de nombre jazzy en el dominio local no tiene acceso al ftp.

La última línea en este ejemplo describe los privilegios de acceso para el servicio de telnet. En este caso, es la subred 192.168.2 ha sido declarada especificamente con acceso al telnet.

Otra característica interesante del tcpd es la que cuando una entrada es encontrada en uno de los archivos hosts.allow o hosts.deny, un comando puede ser ejecutado después que el servicio permitido o denegado. Este asa el campo comando opción. El usuario puede colocar un comando en este campo y este se ejecutará siempre y cuando yna petición iguale esta entrada. Por ejemplo, se puede ejecutar finger para investigar quien desea acceder el servicio y los resultados pueden ser enviados por e-mail a root. Muy a menudo los ataques al sistema pueden ser evitados con simplemente efectuando un monitoreo de quien quiere accesar cuales partes del sistema.

## DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

El DHCP es usado para asignar direcciones IP a los hosts de una red. Las asignaciones son administradas por un servidor centralizado, asi simplificando la administración. En esta sección discutiremos el protocol DHCP y dos de su antecesores, BootP y RARP. También cubriremos el uso y la configuración de los clientes y servidores DHCP sobre plataforma GNU/Linux.

Hay varias buenas razones para usar un DHCP. Una de las principales razones para el uso de un servidor DHCP es para que usted no tenga que configurar los valores de IP de cada máquina manualmente en su red. Este también elimina la necesidad de tener que contabilizar el IP individual asignada a cada máquina. También nos permite tener más máquinas o hosts que direcciones IP disponibles. Esto es muy útil para los ISP o comapñías con muchos usuarios, especialmente los que usan portátiles. Aunque podemos tener más hosts que direcciones IP, usted no podrá exceder el número de direcciones IP con los hosts conectados.

Los siguientes tópicos son discutidos en esta sección:

- Configuración Automatica vs. Dinámica
- RARP
- BootP
- DHP
- Configurar un Servidor DHCP
- Configurar un Cliente DHCP

***Configuración Automática vs. Dinámica***

Los parámetros del TCP/IP pueden ser configurados en varias maneras diferentes. En su origen, toda la información de configuración era estatica e introducida manualmente. Muchos istemas aún son configurados manualmente, especialmente los sistemas que son servidores. RARP y BootP introdujeron la configuración automática. El DHCP permite las dos opciones de configuración automática o dinámica.

En la configuración autmática, el cliente consulta a un servidor para recibir sus parámetros cuando primero se inicia y se une a la red. Estos parámetros, incluyendo la dirección IP, son los mismos todo el tiempo, al menos que el administrador efectúa un cambios en el servidor.

La Configuración Dinámica difiere de la configuración aumática en que al cliente se le asigna una dirección IP y configuración cada vez que se une a una red. A diferencia de la configuración automática, puede ser que se le asignen parámetros diferentes cada vez que inicie el sistema. La dirección IP es alquilada al cliente. El alquiler de una dirección IP expira después de un intervalo de tiempo especificado, luego del cual deberá ser renovado por el cliente o retornado al pool de direcciones y configuraciones. La Configuración Dinámica fué el cambio principal del BootP al DHCP.

La diferencia entre la configuración automática y dinámica es como el servidor determina la información de la configuración. Si este usa un pool de direcciones IP y escoje una al azar, entonces estamos utilizando una configuración dinámica. Si esta usa una tabla creada por el administrador de sistema para mapear direcciones MAC a direcciones IP, entonces estamos efectuando configuración automática. Ambos metódos utilizan un servidor para hacer la asignación actual a los clientes.

El DHCP ofrece tres diferente tipos de configuraciones:

| | |
|---|---|
| **Manual** | Esta es equivalente al BootP. |
| **Automática** | El cliente es automáticamente asignado una dirección IP permanete y su configuración desde que se une a la red. |
| Dinámica | El cliente es asignado una dirección IP y su configuración cada vez que se une a la red. La dirección IP es alquilada al cliente. |

En configuraciones automática y dinámica, el alquiler de la dirección IP (y la configuración del cliente) expiran después del intervalo de tiempo establecido, luego este cliente debe renovar su alquiler y la dirección será retornada al pool de direcciones y configuraciones del servidor.

**RARP**

El primer protocolo que fué usado para asignar direcciones IP automáticamente fué RAPR (Reverse Address Resolution Protocol). El RARP está definido en el RFC 903. Este se encuentra en la misma capa que el ARP y usa algunos de los mismos formatos de paquetes. Así que el RARP se encuentra al nivel inferior (o paralelo) al de la capa del protocolo IP. Esto tiene mucho sentido ya que es imposible enviarle un paquete IP a un cliente que aún no tiene una dirección IP asignada.

El protocolo RARP no usa IP, asi que este debe hacer interface directa con la capa de Enlace (Ethernet Data Link layer). Por esta razón es que el kernel debe implementar este protocolo. Al configurar el kernel Linux, usted notará dos entradas preguntandole si desea habilitar soporte de RARP: uno es para habilitar el sistema que funcione como un cliente de RARP y el otro es para que la máquina funcione como un servidor RARP. El servidor RARP contiene una tabla del mapeado de las direcciones MAC a direcciones IP. Si recordamos cuando cubrimos el ARP, esta tabla mapea las direcciones en forma inversa a la tabla de búsqueda del ARP.

Cuando un cliente de RARP se inicia y desea determinar su dirección IP, este crea un paquete de petición y envia un mensaje de difusión o broadcast atraves de la capa completa de Enlace de Data del Ethernet (Ethernet Data Link layer). Cuando el servidor RARP escucha la petición RARP, este busca la dirección MAC en su tabla de RARP. Si el servidor encuentra la dirección MAC en la tabla, este procede a enviar una respuesta RARP al cliente con la dirección IP que este encontró. Este envia esta respuesta RARP en un paquete raw de Ethernet directamente a la dirección MAC del cliente.

Una de las desventajas de RARP es que este requiere solamente un broadcast desde el cliente y una transmisión unicast desde el servidor al cliente. Pero, existen muchas limitaciones con el RARP, entre las cuales se incluyen:

- Para usar RARP sse requiere acceso directo a la capa de Enlace de Data. Esto hace que la programación de aplicaciones sea un poco más dificil y a menudo el protocolo termina siendo implementado en el kernel. Esta implementación es preferible en el espacio del usuario o el user space.
- Las respuestas RARP solo contienen una única dirección IP, no distribuyen ninguna información acerca de los routers, imagenes de boot del servidor, etc. Otro método es necesario pra proveer la información acerca de las imagenes de arranque y los otros parámetros para los clientes sin discos (diskless clients).
- Como se encuentra en la capa de Enlace de Data, RARP no se desplaza más allá de la red local. Por esto es que debe existir un servidor RARP en el mismo segmento local de cada cliente sin disco.

Para sobre pasar algunos de estos problemas, se desarrollaron estos protocolos de BootP y el DHCP.

**BootP**

Como el RARP, el protocolo BootP provee a los sitemas con direcciones IP que son automáticamente configuradas, pero provee más informacióny utiliza diferentes mecanismo. Además de informar sobre la dirección IP de un cliente, el BootP puede también proveer mucha más información. Algunos ejemplos son: la pasarela local por defecto (default gateway), direcciones de servidor de nombres, la dirección IP de servidor de imagen y el nombre del archivo de la imagen boot. El BootP fué descrito en el RFC 951 y fué disennado para ser usado por estaciones de trabajo sin discos. Para los clientes sin discos, el boot ROM obtiene la información del BootP y usa la información para contactar el servidor TFTP para cargar el sistema operativo.

Cuando un cliente BootP desea obtener una dirección IP, este envia un broadcasts de un paquete con un BOOTREQUEST a la subred local. Esto se logra usando la dirección IP de broadcast, 255.255.255.255. Por lo general la dirección IP será 0.0.0.0 en este paquete, indicando que el paquete proviene de un sisstema desconocido o uno sin una dirección IP válida. El broadcast es enviado al puerto UDP número 67. La respuesta es enviada al puerto UDP número 68, asi pues que el cliente debe escuchar en ese puerto por los paquetes de respuesta.

Al igual que el RARP, el servidor BootP busca en sus tablas para determinar a cual dirección IP enviar devuelta a los clientes. Este también puede contener información adicional que desea enviar de vuelta al cliente. Cada pedazo de información enviada es asociado con una etiqueta. La etiqueta describe el significdo

de la información. Las etiquetas son identificada en el documento RFC que describen el protocolo. El tipo de información que usted configura al servidor que efectué depende de las necesidades del sistema cliente involucrado. Por ejemplo, una terminal sin disco puede ser enviada la dirección IP de un servidor TFTP, mientras que a otro sistema le enviariamos otro tipo de información.

Para enviar el paquete de respuesta al cliente puede que resulte un poco complicado para el servidor ya que este debe enciar un paquete IP a un sistema sin dirección IP. Hay cuatros posibles métodos que pueden ser utilizados para enviar el paquete al cliente:

- Si el cliente requiere una dirección IP específica en el paquete de requiriente y el servidor concede al cliente usar esa dirección IP, la respuesta puede ser enviada directamente a la dirección IP del cliente.
- Enviar el paquete de respuesta vía un broadcast. La petición que el cliente envia contiene un identificar únco, asi que el cliente deberá escuchar en su puerto UDP número 68 por un paquete de respuesta que iguala su identificador.
- Usar la información del paquete que el cliente envia para forjar una entrada en la tabla de ARP del servidor. Entonces se puede enviar el paquete a la dirección IP y la pila de IP del servidor usará la entrada en la tabla ARP para mapear la dirección IP a su dirección MAC. El kernel Linux soporta este método. De hecho, se puede agregar una entrada a la tabla ARP manualmente usando el comando arp.
- Crea un paquete data-link por completo para enviar sobre el cable de la red. Esto además requiere el soporte del kernel y no es soportado muy bien. El servidor BootP deberá saber todo lo interno del IP asi como los detalles de la capa de Enlace de Datos.

Aunque el BootP es más poderoso que el RARP, si tiene algunas limitaciones. Este fué diseñado para un ambiente de redes estático y asi que los parámetros BootP de cada host son almacenados en un archivo en el servidor BootP. Cambios efectuados a la red requieren la edición del archivo de configuración del servidor. El BootP no puede manejar la asignación de parámetros dinámicamente. El DHCP si fué desarrollado para trabajar en ambientes dinámicos más modernos.

**DHCP**

El DHCP es una extensión del BootP. Su mejora principal son su gran número de parámetros de configuración que pueden ser pasados a los clientes, un método para que el cliente descubra información a cerca del servidor y la habilidad de asignar direcciones IP dinámicamente desde un pool. La versión actual del DHCP está especificada en los RFCs 2131 y el 2132.

Una convesación típica de un DHCP se describe en los siguientes pasos:

1. Primero una máquina “A” envia una trama de Descubrir-DHCP.
2. El servidor B (y puede ser otros servidores) responde con una Oferta-DHCP.
3. El Cliente “A” requiere una configuración al servidor “B” con una Petición-DHCP.
4. El servidor “B” envia la configuración con un Paquete-DHC.
5. El Cliente “A” ahora puede configurarse asi mismo.

Los pasos 3 y el 4 son los mismo como los de un BootP. Pasos 1 y el 2 son una extensión que permiten que el cliente encuentre investigue que servicios DHCP están disponible. En esta manera, el servidor DHCP puede servir a clientes BootP asi como a los clientes de un DHCP.

***Reconectarse a Otra Red***

Después de reiniciar, “A” trata de usar su configuración existente (DHCPRequest). “C” envia un DHCPNAck; esa configuración es inapropiada. Una sesión nueva DHCP se inicia. Ahora el cliente que fué previamente configurado vía el DHCP es conectada a una red diferente.

"A" envia su dirección IP anterior como parte del DHCPRequest. "B" puede entonces examinar esto (y otros detalles de configuración) para decidir si acepta la petición de "A" y le permite usar esa configuración.

Un proceso similar ocurre con la configuración dinámica cuando el alquiler de un cliente ha expirado. El cliente efectuá un DHCPRequest para ver si aún puede seguir usando su configuración existente.

Si un cliente desea desconectarse de la red y liberar su configuración, este puede enviar un DHCPRelease a su servidor. En esta manera, su dirección IP se incluye como disponible en el pool de direcciones sin asi tener que esperar que el alquiler expire.

Fijese que, al igual que con el BootP, existen identificadores de números enteros pasados en cada mensaje DHCP para que asi múltiples clientes y servidores puedan mantenerse informado el uno del otro.

Una limitación de DHCP es que no hay mecanismo alguno para interactuar con el DNS (Domain Name System), el cual mapea nombres de computadoras a direcciones IP. Las nuevas direcciones alquiladas no se agregan al DNS. Algunas implementaciones de DHCP pueden incluir cierto métodos de actualizar el DNS, pero en la actualidad no hay un método estándarizado para esto. Un nuevo sistema llamado DDNS (Dynamic DNS) está siendo desarrollado para sobre pasar estas limitaciones, agregar la información a la base de datos del DNS de manera que se le pasa al cliente. Como esta es la limitación principal, en el presente, del DHCP, se espera un rápido desarrollo en el futuro cercano.

***Relay Agents***

Ambos el BootP y el DHCP permiten al servidor residir en una red no local a traves del uso del concepto de un agente de relevo (relay agent). El relay agent pasa o reenvia las peticiones de los clientes desde una red al servidor y reenvia la respuesta también. No hay problema con confundir diferente clientes ya que el identificador de las direcciones de la Capa de Enlace es única.

Los enrutadores por lo general no reenvia o pasan paquetes de broadcast. Así que debe existir algún método que permita al servidor DHCP escuchar las peticiones desde otras subredes. Hay dos métodos para llevar esto acabo. Podemos instalar un DHCP relay agent (agente de relevo de DHCP) en cada subred que tiene clientes de DHCP o usar un enrutador que pueda ser configurado para pasar el broadcast en el servidor DHCP.

Otro método posible es configurar los enrutadores entre los segmentos. Lo configuramos para que reenvien los broadcasts al servidor DHCP/BootP al puerto UDP (puerto 67). Dependiendo en el enrutador, podemos tener el enrutador reenviar la petición directamente al servidor DHCP o generar otro broadcast.

***Configurar un Servidor DHCP***

La ISC (Internet Software Consortium) ha desarrollado una referencia de la implementaci´øn de cliente y servidor DHCP. La podemos encontrar en su página web http://www.isc.org/products/dhcp. En está sección cubriremos la configuración de un servidor DHCP.

Si usted compiló su kernel, será necesario que se asegure que incluyó el soporte de las operaciones multicast. La mayoría de los kernels que se incluyen en las distribuciones estándares traen este soporte habilitado. Podemos revisar si nuestro kernel lo tiene habilitado, observando la salida del comando ifconfig para observar si se incluye la palabra MULTICAST:

```
# ifconfig  eth0
eth0    Link encap:10Mbps   Ethernet    HWaddr  00:CO:1F:12:23:AB
inet addr:192.168.2.10   Beast:192.168.2.255 Mask:255.255.255.0   BROADCAST   RUNNING  MULTICAST
MTU: 1500    Metric:!
```

Si tiene clientes de otros sistemas operativos aceptando servicios DHCP en la red, puede que deberá eje-

cutar el siguiente comando:

```
#    route add   -host  255.255.255.255  dev  eth0
```

Si es necesario entonces, tendrá que agregarlo a los scripts de inicio, quizas en el archivo /etc/init.d/rc.local.

***dhcpd***

El daemon servidor DCHP es llamado dhcpd. Por lo general es instalado en el directoio /usr/sbin, con un script de inicio en el directorio /etc/rc.d/init.d. Si su servidor DHCP es multihomed o actuá como un enrutador, tendrá que decirle al servidor DHCP en cual interface de red servir. Por ejemplo, si deseamos que sirva la red conectada a la primera tarjeta de red, deberá ejecutar la siguiente sentencia:

```
#  /usr/sbin/dhcpd   eth0
```

Antes de iniciar el dhcpd, asegurece de que el archivo /etc/dhcpd.leases existe. Puede ser que diferentes distribuciones coloquen este archivo en otro directorio, como puede ser /var/state/dhcp/dhcpd.leases. usted puede lograr esto invocando el siguiente comando:

```
# touch  /etc/dhcpd.leases
```

Mientras pone a prueba el servidor DHCP, podemos ejecutar con debugging (modo de depuración de errores) habilitado y no ejecutarlo en el segundo plano o background, invocandolo de la siguiente manera y con las opciones aqui mostradas:

```
# /usr/sbin/dhcpd   -d    -f
```

Ejecutelo en modo de debug, observando los mensajes al inicio de uno de los clientes de DHCP. Después de asegurarse de que los clientes están recibiendo si direcciones IP del servidor, agregue el daemon servidor a los scripts de inicio.

***dhcpd.conf***

El archivo de configuración principal del dhcpd es /etc/dhcpd.conf. Existen interfaces gráficas para editar este archivo de configuración, unas de ella es llamada kcmdhcpd, pero como el archivo es de texto plano tipo ASCII, lo más probable es que usará un editor de texto para configurarlo. La configuración tiene parámetros globales asi como secciones para establecer parámetros específicos a una subred o máquina.

Existen muchas opciones que pueden ser especificadas para ser enviadas al cliente y varias maneras de específicar que debe ser enviado a cada cliente. Aquí le presentamos un archivo dhcpd.conf de ejemplo:

```
default-lease-time  3600;
option  subnet-mask  255.255.255.0;
option   routers  192.168.2.1;
option  domain name-servers  192.168.2.1,   192.168.3.1;
option  domain-name   “mi-dominio.org”;

subnet 192.168.2.1  netmask  255.255.255.0  {
     range  192.168.2.10  192.168.2.100;
     option  netbios-name-servers  192.168.12.21;
}
host miservidor  {
     hardware  Ethernet  00:0C:12:43:AF:BE;
     fixed-address   192.168.2.13;
     option  routers   192.168.2.1  192.168.2.2;
}
```

La primera línea establece el alquiler para los clientes a 3,600 segundos o 1 hora. La sentencia “option” específica la información que va a ser enviada a los clientes. Las primeras cuatro opciones son enviadas a los clientes DHCP.

La sección que inicia con “subnet” dice que cualquier cliente que este en el segmento 192.168.2.0 debe ser asignado una dirección IP a azar desde el pool de direcciones 192.168.2.10 al 192.168.2.100. Observe que la opción netmask es usada para especificar el tamaño de la subred que está definido encerrado entre las llaves, donde la opción subnet-mask específica la información que será enviada al cliente. La opción netbios-name-servers establece el servidor de nombre para los clientes de otro sistema operativo.

La sección que inicia con “host” muestra como establecer los parámetros para un host en específico. Usando este método, podemos especificar la dirección de hardware o MAC del sistema. recuerde que esto significa que si el NIC del sistema es cambiado, usted tendrá que reeditar el archivo de configuración del DHCP server. En este cso, específicamos que el sistema con la dirección MAC de valor 00:0C:12:43:AF:BE debe ser asignada la dirección IP número 192.168.2.13. La opción específicada en la sección host-specific anteceden en preferencia a las opciones en la sección global.

**Configurar un Cliente DHCP**

Básicamente todos los clientes DHCP efectuán la mismas tareas en diferentes formas. Ellos permiten que hosts obtengan una dirección IP dinámicamente cada vez que se efectuá un login o ingreso al sistema. Este protocolo es extremadamente importante en las redes amplias y complejas de hoy día. Usando el DHCP, múltiples hosts pueden ser administrados facilmente por el administrador atraves de una base de datos central sin tener que ejecutar un gran número de configuraciones a equipos individuales. El DHCP también permite que redes puedan acomodar un número de hosts mayor al número de direcciones IP disponibles, previendo, claro que no todos los hosts se conecten a la misma vez. Esta técnica es muy usada por los grandes ISPs que tienen una amplia matricula de subscritores con sólo un número limitado de direcciones IP que poseen para asignar. Ellos asumen que no todos los usuarios ingresarán al sistema simultáneamente. El DHCP también permite que un gran número de usarios moviles sean asignados una dirección y que se comuniquen sin importar que conexión física los une a la red, sin tener que cambiar su configuración.

Existen tres clientes principales para el protocolo DHCP: pump, dhcpcd y dhclient. Estos programas clientes interactuán con un servidor DHCP para obtener la información de configuración de la red. Ellos se ejecutan como daemons, siempre revisando para asegurarse que el alquiler aún es válido y renovarlo si es necesario. Si la conexión al servidor se detiene, entonces el programa cliente continuará tratando de contactar el servidor hasta que la conexión sea restaurada.

***pump***

El cliente pump es un paquete simple de DHCP desarrollado por Red Hat. Es el cliente más fácil de usar e incorpora las mismas características que otros paquetes mucho más difíciles de implementar. Se inicia automáticamente por el script /sbin/ifup pero puede ser ejecutado manualmente para revisar la configuración o el estado de la conexión. El comando pump tiene un número de opciones, pero la sintaxis de la línea de comandos es bien simple:

```
# /sbin/pump   -option
```

Esta es una lista de las opciones básicas:

| | |
|---|---|
| -c | Define archivo diferente de configuración y no el /etc/pump.conf |
| -h | Nombre de host que se requiere. |
| -i | La interfaz a ser configurada (eth0, eth1, etc.) |
| -k | Mata/Kill el programa (el daemon y todas las interfaz) |
| -l | Tiempo de alquiler (horas) |

| | |
|---|---|
| -R | Renovación inmediata del alquiler al expirar |
| -s | Estado de la interfaz |
| -d | Detiene el cambio del DNS (sin cambios al /etc/resolv.conf) |
| -? | Help/Ayuda |
| --usage | Uso de la ayuda |

Usted puede revisar el estado del daemon pump escribiendo:

```
# /sbin/pump -i eth0
```

La salida sería algo parecido al siguiente ejemplo:

```
Device eth0
IP: 192.168.2.10
Netmask: 255.255.255.0
Broadcast: 192.168.2.255
Network: 192.168.2.0
Boot server 192.168.2.4
Next server 192.168.2.4
Gateway: 192.168.2.7
Domain: codigolibre.org
Nameservers: 192.168.5.54 192.168.5.254
Renewal time: Sun Jun 16 23:54:42 2004
Expiration time: Mon Jun 23 00:09:42 2004
```

Este comando retorna toda la información que fué asignada al cliente por el servidor DHCP. Aquí el usuario puede revisar cuando el alquiler expira y cuando el pump renovará el alquiler.

El pump mantiene la información de su configuración en el archivo /etc/pump.conf. Este es un archivo orientado a líneas que contiene comandos y argumentos parecidos a los comandos del shell, que incluye comillas y barras invertidas (\). Hay dos maneras diferentes de configurar un sistema: global o específica. La global afecta toda todas las conexiones manejadas por un daemon, mientras que la específica maneja conexiones únicas. Al definir el dispositivo con las opciones, los cambios pueden ser específicas a ese dispositivo; de cualquier otra manera, los cambios se efectuarán en todo el sistema. Aquí le presentamos un archivo de config de ejemplo que define un dominio para efectuar búsqueda, el número de reintentos a efectuar, y el dispositivo a configurar:

```
domainsearch "codigolibre.org abiertos.org"
retries 1
device eth2 {nodns}
```

Este archivo de configuración permite que cierto aspectos puedan ser cambiados con el comando pump desde la línea de comandos para ser integrados como constante en los arranques subsecuentes.

***dhcpcd***

El dhcpcd es un cliente DHCP poderoso que se puede usar con cualquier distribución de GNU/Linux. Este efectuá el mismo trabajo que el pump pero puede ser más difícil de usar. El dhcpcd tiene una línea de comadnos un poco más compleja debido al alto número de opciones necesarias para configurar una cosa, pump fué hecho para el usuario principiante, mientras que configurar con el dhcpcd es mejor dejarselo a los usuarios experimentados.

La línea de comando, dhcpcd [-opción argumento] [interfaz], se ve simple, pero entrar en la opciones toma cierto tiempo. Aquí es una pequeña lista de opciones disponibles y sus próposito:

| | |
|---|---|
| d-d | Escribe al log syslog con mucho detalle. |
| -k | Mata/Kills la conexión actual y elimina el cache (permite nuevas peticiones de direcciones). |
| -B | Hace una petición de una respuesta tipo broadcast desde el servidor. |
| -D | Obliga al dhcpcd a usar el nombre de dominio retornado por el servidor. |
| -H | Obliga al dhcpcd usar el nombre del host retornado por el servidor. |
| -R | No cambiar el DNS; no reemplazará el archivo /etc/resolv.conf. |
| -t | Timeout/Tiempo agotado (en segundos). |
| -c filename | Ejecute el archivo de nombre filename (por lo general un script). |
| -h hostname | Específica un nombre de host que le enviará al servidor (como un login). |
| -i VTD | ID Vendor. |
| -I CID | ID Cliente. |
| -l | Envia el tiempo de alquiler recomendado (en segundos). |
| -s IP | Hace la petición de una dirección IP específica; si se deja en blanco, devolverá la IP actual. |
| -s IP interface | Define la interfaz a ser configurada. |

El dhcpcd utiliza múltiple archivos para contener la información que ha capturado. La mayoría de estos están localizados en el directorio /etc/dhcpcd. La parte <interface> de los archivos es reemplazada con el dispositivo que usted está usando. Se produce un archivo por cada dispositivo que va a ser configurado. Aquí en lo adelante se listan los archivos necesarios y el propósito de cada uno.

**/etc/dhcpcd/dhcpcd-<interface>.info**

Este contiene la información que el dhcpcd ha adquirido; es usado para mantener los mismos parámetros cada vez si e posible.

**/etc/dhcpcd/dhcpcd-<interface>.exe**

Este se ejecuta cada vez que es cambiado el IP del sistema; este actualiza toda la información del sistema.

**/etc/dhcpcd/dhcpcd-<interface>.pid**

Este contiene la ID del proceso requerido por el servidor.

**/etc/dhcpcd/dhcpcd-<interface>.cache**

Este contiene la dirección previamente asignada usada por el dhcpcd para readquirir la misma dirección en cada conexión; esta puede ser limpiada co la opción -k.

**/etc/resolv.conf**

Este archivo contiene las opciones de nombre de dominio y DNS retornada por el servidor; es limpiado cada vez que una nueva conexión es efectuada al menos que la opción -R es usada.

Es importante señalar que no debe confundir el daemon servidor, dhcpd, y el daemon cliente, dbcpcd.

***dhclient***

El dhclient es un cliente DHCP mucho más simple para las distribuciones que no soportan pump. El dhclient utiliza un archivo de configuración simple en vez de de cambiar la configuración desde la línea de comandos. la línea de comandos es dhclient -option con -p port para especificar el puerto que el dhclient transmite y -d para ejecutar el dhclient en modo de depuración o debug mode.

El dhclient revisa el archivo /etc/dhclient.conf antes de inicializar las instrucciones de configuración. El archivo contiene toda la información necesaria por el dhclient y la información adquirida dsde el servidor DHCP. Aquí le presentamos un ejemplo del archivo dhclient.conf:

```
$ dhclient.conf
timeout   60;
```

```
retry 60;
reboot  10;
select-timeout  5;
initial-interval   2;
interface  “eth0”   {
     request  subnet-mask,   broadcast-address, time-offset,
     routers,  dhcp-lease-time,
     domain-name,  domain-name-servers, host-name;
     require subnet-mask,  domain-name-servers;
     send dhcp-lease-time  24000 ;
}
```

El archivo es similar al archivo de configuración de pump. Podemos específicar opciones globales o específicas, como se demuestra aquí con la opción interface. Fijese, como la llaves {} deben ser usadas para asignar opciones asociadas con una interfaz, el dhclient también necesita un archivo dhclient.leases. Este archivo contiene toda la información de todas las direcciones IP que han sido asignadas, el dhclient agregará nuevas entradas al final del archivo. Pero estas adiciones pueden causar un problema. Listas masivas de alquiler que consisten en miles de viejos alquileres toman mucho espacio de disco, el dhclient resuelve esta problematica crando un nuevo archivo de alquiler periodicamente, dejando los alquileres anteriores en archivos por separados de resguardo/backup. Estos archivos de backup pueden entonces ser eliminados si es necesario o retenidos para cuestion de record.

### Ejercicios 7-2: Configurar el DHCP

En este ejercicio se rquieren dos computadoras conectadas en red. Instalaremos un servidor DHCP en una de ellas. Necesitaremos un cliente en la otra para hacer la operación de prueba. En este ejercicio deberá usar la dirección de subred privada 192.168.0.0. No se proveen soluciones para este ejercicio.

1. Instale el dhcpd. En muchas distribuciones, ya este estará instalado. Si no es el caso, es recomendable que usted instale el paquete proveido por su vendor. Alternativamente puedes descargar el fuente desde el ftp.isc.org descomprimirlo, desempaquetarlo, compilarlo e nstalarlo.

2. Determine donde se encuentra su archivo dhcpd.leases. Puede que este en /etc o quizas en /var, como en /var/state/dhcp. Las páginas man de su paquete le puede ayudar a ubicar su archivo dhcpd.leases:
   **# man 5 dhcpd.leases**

3. Si el archivo dhcpd.leases no existe deberá crearlo:
   **# touch  /etc/dhcpd.lease**

4. Edite el archivo /etc/dhcpd.conf para que contengan las siguientes líneas:

```
dcfault-lease-time 3600;
option  subnet-mask  255.255.255.0;
option  routers  192.168.0.1;
option domain-name-servers  192.168.0.1,   192.168.0.9;
option domain-name  “mi-dominio.org”;

subnet  192.168.0.0  netmask  255.255.255.0 {
     range  192.168.1.20   192.168.1.250;
     option  netbios-name-servers   192.168.0.15;
}
```

5. Inicie el Servidor DHCP en modo de depuración (debug):
   **# /usr/sbin/dhcpd   -d   -f**

6. Inicie el cliente DHCP. En el sistema que debe estar configurado para usar DHCP y luego podemos revi-

sar para ver que parámetros nos asigno.

7. Observe la salida de depuración que el daemon servidor dhcpd expide.

8. Revise los parámetros que el cliente DHCP recibió. En GNU/Linux, con en la mayoría de sistemas operativos podemos usar los comandos ifconfig y route sin argumento alguno.

9. Renove el alquiler de la dirección que recibió. Con el cliente de pump, usted puede efectuar esta tarea escribiendo la siguiente sentencia:
   **# pump --renew**

10. Verifique si puede renovar su alquiler de DHCP y recibir uno nuevo. Con el cliente pump, usted puede lograr esta tarea primero relegando el alquiler y luego contratar uno nuevo:
    **# pump --release**
    **# pump --renew**

    Con el dhcpcd, tendrá que matar el daemon, deshabilitar la interfaz y rehabilitarla.

## TELNET

El servicio de Telnet es uno de lo más usado en los sitemas UNiX y lo más seguro es que usted ya este familiarizado con este. Telnet provee acceso a una conexión de terminal virtual en una red. Esta conexión virtual posibilita ejecutar programas de consola igual que si estubiese sentado en el equipo remoto. La naturaleza del uso de la línea de comandos de GNU/Linux y UNiX apodera al administrador de sistema manejar su sistema de cualquier ordenador y de cualuier punto en la red.

El servidor telnet escucha por conexiones entrantes en el puerto TCP 23. El servidor acepta conexiones desde los clientes, luego inicia un proceso de login a esa conexión. En GNU/Linux, el servidor utiliza una técnica conocida como “pseudo terminal” para simular un terminal físico. Los pseudo terminales son dispositivos de software que simulan el comportamiento de un terminal, aunque no esten conectadas a un dispositivo físico sino a algún tipo enlace de comunicación. Ellas forman la base de la gran mayoría del software de emulación remota en GNU/Linux. Hasta en los emuladores de terminales locales como es el xterm se emplea el uso de pseudo terminales. Un pseudo terminal es mejor conocido como un pty.

La comunicación de los caracteres entre el cliente y el servidor se lleva acabo utilizando un formato independiente de máquina conocido como NVT (Network Virtual Terminal). Es responsabilidad del cliente y el servidor en cada punta convertir entre el formato NVT y las caracteriísticas del teminal del sistema local.

El telnet es asignado un único canal para los comandos y la data por definición del RFC 854. Telnet permite que los clientes inicien y tomen parte de sesiones interactivas en sistemas remotos. El equipo cliente es capaz de iniciar sesión en el servidor para asi poder transferir data y llevar acabo otras actividades interactivas.

El servidor daemon de telnet es por lo general ejecutadod esde inted, asi que usted puede usar los wrappers de TCP para filtar acceso. Su ejecutable es nombrado in.telnetd o telnetd. The following represents the entry chat you should use in /etc/inetd.conf using TCP wrappers:

```
telnet  stream  tcp  nowait  root  /usr/bin/tcpd  in.telnetd
```

Una característica interesante del programa telnet es que usted puede ejecutar un telnet a cualquier puerto TCP. Cuando hacemos esto, usted termina comunicandose con el servicio de red y por ende debe usar los comandos raw definidos por el servicio. De hecho, no tendrá ningún software cliente y deberá actuar como

el programa cliente mismo. En el siguiente ejemplo ilustramos una conexión a un sistema FTP directamente:

```
$ telnet ftp.abiertos.org 21
Trying 192.168.2.15...
Connected to ftp.abiertos.org.
Escape character is '^]'.
220 ftp.abieros.org FTP server (Version 2.6.0) ready,
user anonymous
331 Guest login ok. send your complete e-mail address as password.
pass mike@abiertos.org
230-Welcome to My FTP Site.
230 Guest login ok. access restrictions apply.
cwd /pub
250 CWD command successful .
pwd
257 "/pub" is current directory.
type i
200 Type set to I.
size archivo.txt
213 1416
quit
221 Goodbye.
```

En esta sección, discutiremos tópicos que concernientes de seguridad.

***Seguridad y Telnet***

Telnet no debe ser usada en conexiones abiertas atraves del Internet. Cualquier contrasenna que escriba puede ser interceptada ya que telnet no las encripta. Este las envia por la red en formato de texto plano. Para el acceso seguro de un shell, deberá usar el programa SSH (Secure Shell).

El SSH provee encriptación entre el cliente y el servidor. Este método de transferencia encriptada protege tanto las contraseñas como la información que se transmite entre los dos sistemas. Sin tomar en cuenta esta seguridad agregada, el SSH funciona igual que el telnet.

Hasta en un ambiente corporativo, usted debiera considerar utilizar los servicios del SSH. Como el Ethernet es un medio compartido, cualquier sistema en cualquier segmento que la data atraviesa puede escuchar a todo que pasa atraves de el. El Token Ring es también un medio compartido. Nunca se puede estar seguro de quien está huzmeando nuestro tráfico. Recuerde que un gran porcentaje de los ataques a los sistemas son cometidos desde adentro. Utilizar switches en su red agrega un nivel mayor de de seguridad ya que solamente direcciona paquetes al destino final. Aún asi, la data deberá viajar atraves de enrutadores y hasta redes desconocidas.

**SSH/OPENSSH**

Es obvio la necesidad de inclementar los nivels de seguridad entre las redes a medida que que más y más personas utilizan el Internet para intercomunicarse. En el pasado, los métodos de comunicación con aplicaciones como es el telnet no ofrecen niveles aceptables de seguridad. Cuando un usuario se conecta remotamente, existe el potencial de ser monitoreado por una fuente externa. Esta fuente externa puede ser un cracker o cualquiera intentado ingresar ilegalmente a nuestra red. La seguridad para los acceso remotos debe ser aplicado igualmente a los usuarios del hogar y los corporativos en su uso del Internet. Para contrarestar la violaciones de la red, un método ha sido desarrollado para proveer niveles de seguridad más alto para los ingresos interactivos remotos llamado Secure Shell (SSH).

Existen dos versiones del SSH. La versión 1 fué lanzada de licenciamiento libre y es mantenida como el

OpenSSH. La versión 2 opera bajo protocolo diferente y es sólo libre para su uso no comercial. Desde el punto de viasta del usuario, ambas operan de la misma manera y con los mismos resultados. El comando ssh lanza ambas versiones.

El usuario utilizza el programa de generar llaves ssh-keygen para generarse un par de llaves privada y pública, luego coloca la llave pública en el host remoto en un de dos directorios $HOME/.ssh/authorized_keys para el SSHl o el directorio $HOME/.ssh2/authorization para el SSH2. Una vez la llave pública ha sido colocada en el host remoto, los usuarios locales pueden ingresar remotamente al host usando la clave (passphrase) secreta del SSH.

Los siguientes tópicos son discutidos en esta sección:
- SSHl
- SSH2
- OpenSSH y Stunnel
- Instalación
- Configuración
- Usado

**SSH1**

El SSH1 es la primera versión del programa Secure Shell y es usada para ingresar remotamente a un sistema. Es la manera segura de obtener una sesión de login interactiva segura. La función principal del SSH1 es la de encriptar cada paquete que atraviesa desde una máquina hacia una red externa. Esto previene que los crackers puedan decodificar los paquetes después de interceptarlos. Si un cracker logra interceptar nuestros paquetes, la desencriptación de nuestro paquete será una ardua tarea. Asi que, el chance de que nuestra información sea explotada exitosamente son muy pocos. En resumen, usar el SSH al conectarse remotamente a una red protejerá su información mucho más seguro que usando telnet y cualquier otro cliente. El programa SSH1 soporta cuatro diferente tipos de cifrados para su encriptación: DES (Data Encryption Standard), 3DES (Triple DES), IDEA (International Data Encryption Algorithm), y Blowfish. El cifrado soportado para autentificación es RSA, que son las consonantes de los nombres Rivest, Shamir y Adleman del cifrado de encriptación. El SSH1 ya no es desarrollado en la actualidad.

**SSH2**

La nueva versión del programa Secure Shell es el SSH2, el cual ofrece una mejor estructura para el protocolo. La implementación básica del SSH2 es completamente diferente a la del SSH1. Como mencionamos anteriormente, el SSH encripta los paquetes de data antes de enviarlo y los desencripta en el host remoto. El SSHl y SSH2 son distintos en que ellos encriptan la data en doferentes puntos del paquete, haciendo la encriptación diferente en estructura y composición. El SSH2 posee una encriptación más altaque provee mejor seguridad, mejor y más confiable rendimiento y esta disponible en varias ediciones y para una variedad de sistemas operativos. Estas nuevas características hacen que el SSH2 sea un estándar de seguridad deseado. El SSH2 tiene 5 cifrados para encriptar: 3DES, Blowfish, Twofish, Arcfour y Cast 128-cbc. El cifrado soportado para la autentificación es el RSA (Digital Signature Algorithm).

***OpenSSH y Stunnel***

OpenSSH es un paquete de programas que contienen un conjunto de herramientas de red útil para la implementación del protocolo SSH. Entre estas herramientas se incluyen el programa SSH y los comandos sshd, ssh-agent,ssli-add,scp y ssh-keygen. Estos comandos serán brevemente descripta más adelante:

**sshd**

El Secure Shell Daemon se ejecuta constantemente y escucha por otros hosts que intentan establecer una conexión segu-

ra con SSH.

**ssh-agent**

Es activado cuando uno ingresa a una localidad remota. Este proteje la llave RSA usada en la autentificación.

**ssh-add**

Este comando agrega y actualiza las identidades al agente de autentificación.

**scp**

El coamndo scp oermite la copia segura de archivos desde un host en una red a otro.

**ssh-keygen**

Este genera una llave para que los paquetes tanto entrante como saliente tengan autentificación y sean aceptados y aprovados.

El programa stunnel es similar al SSH en que este encripta los paquetes para una comunicación segura. El stunnel utiliza autentificación SSl (Secure Socket Layer) con las librerías OpenSSL. Una ventaja de este programa ocurre cuando un sistema se encuentra ejecutando un daemon SSL y la computadora que se comunica no está ejecutando SSL. Un canal de comunicación segura puede ser establecida sin importar el conflicto. El stunnel también puede protejer contra la intercepción y la manipulación de data durante la transmisión a su destino primario.

***Instalación***

OpenSSH se incluye con la mayoría de las distribuciones y el fuente puede ser obtenido desde el portal http://www.openssh.com. SSH2 puede ser encontrado en http://www.ssh.com. El SSH original es soportado al igual que el protocolo SSH2 en las versiones 2.3 en adelante del OpenSSH. Después de extraer el archivo .tar, el fuente deberá ser configurado con el comando ./configure. Luego, deberá ser compilado con el comando make. Finalmente, tendrá que instalar el software con el comando make install.

***Configuración***

El servidor lee su configuración desde el archivo /etc/sshd_config y el cliente lo lee desde /etc/ssh_config y entonces desde uno $HOME/.ssh/ssh_config para el SSH o desde el $HOME/.ssh2/ssh2_config para el SSH2. Algunos administradores de sistemas puede que deseen aliviar algunas de las restricciones por defecto editando el archivo sshd_config.

***Uso***

El comando ssh debe ser ejecutado por lo menos con un argumento- el nombre del host al cual deseamos conectarnos. Si ningún otro argumento es suplido, un shell en el host remoto se iniciará, ssh también puede ser ejecutada en el formato:

**# ssh HOST COMANDO**

el cual puede causar que ssh ejecute el COMANDO en el HOST remoto. Si el comando a ser ejecutado es interactivo (por ejemplo, un cliente de ftp o correo), la opción -t debe ser usada (ssh -t HOST COMANDO), o (con el SSH2 solamente) use el la directriz ForcePTTYAllocation en el archivo sshd_config.

El ssh también puede reenviar conexiones X11. Esto permite que un usuario se conecte a programas desde el host remoto a su display X local y con un alto nivel de seguridad a diferencia del VNC. Este debe ser habilitado con la directiva ForwardX11 en el archivo sshd_config.

El comando para ingresar a un sistema que tiene ssh instaldo y ejecutándose es parecido al siguiente:

**$ ssh [ -l login name] hostname [command]**

El parámetro -1 permite al usuario a especificar un nombre para el login. El nombre de host es un parámetro requerido para representar el computador que se desea establecer la conexión. Una lista de opciones más completa se encuentra en la página man del comando SSH.

La aplicación de ssh debe usarse cuando se desea conectar a una localidad remota. Por ejemplo, para

conectarse a codigolibre.org para revisar el espacio en disco disponible, la sentencia ssh sería así:

```
$ ssh  -l  jazmine codigolibre.org
```

En este comando ssh, se muestra como ingresar al sistema codigolibre.org usando el nombre de usuario jazmine. La opción -1 pasa el nombre del usuario para esta sesión, jazmine, al programa. En algunos sistemas, los argumentos se pueden presentar en al reves, asi que esta sentencia también es permitida:

```
$ ssh  codigolibre.org   -l  jazmine
```

Despué que una conexión segura ha sido establecida al host, el servidor le pedirá la contraseña para la cuenta del usuario sometido después de la opción -l.

Al usar una llave pública de autentificación, el usuario puede usar ssh-agent para sólo abrir la llave privada una vez, el ssh-agent es mejor usado para iniciar el shell del usuario o una sesión del X. El administrador puede establecerlo finalizando su script del sistema del arranque del X con la sentencia:

```
exec ssh-agent $HOME/.xsession
```

o el usuario puede habilitarla finalizando su archivo $HOME/.xsession con:

```
exec ssh-agent WINDOW-MANAGER
```

Para que el ssh-agent pueda hacer cualquier cosa, las llaves deberán ser agregadas con el comando ssh-add. Al ejecutar, ssh-add le pedirá la clave o passphrase para desbloquear la llave privadato . Asi que, todos los procesos SSH que son decendientes del proceso ssh-agent process no le pedirán la clave o passphrases ya que la autenficación se ha llevado acabo.

El SSH biene además con el programa lamado scp (Secure Copy). Es usado para hacer copias seguras de archivos desde un host a otro. Aunque es un reemplazo para el ftp, SSH2 también incluye un reemplazo aparte del ftp llamado sftp. El scp es ejecutado así:

```
# scp Fuente Destino
```

Donde Fuente es en el fomato de host:nombre-archivo. También se pueden suplir los nombres de usuarios, por ejemplo:

```
# scp archivo root@codigolibre.org:/etc
```

El sftp es similar al ftp y soporta los mismo comandos básicos del ftp como lo son: ls, cd, put, get, etc.

**FTP**

El FTP es el Protocolo de Transferencia de Archivos (File Transfer Protocol), que provee la transferencia de archivos de texto y binarios. Es uno de los protocolos IP básicos y es casi tan antiguo como el propio TCP/IP. En esta sección, daremos un vistaso a la parte básica del protocolo FTP y el uso básico de tanto el cliente como el servidor FTP. Además cubriremos las implementaciones particular de los servidores wu-ftpd y ProFTPD.

Aunque muchas transferencias de archivos son efectuadas usando HTTP, el FTP continua presentando algunas ventajas. Como por ejemplo, es mucho más fácil resumir el transfer si se detiene y el FTP también mantien los bits de información de los archivos como son los permisos y las fechas de modificaciones. El FTP permite al cliente navegar atraves de un sistema de archivos en el servidor, proveiendo un sistema de archivos virtual que no está disponible en el HTTP. El FTP también puede transferiri múltiples archivos en una sóla operación.

Los siguientes tópico son discutido en esta sección:

- Clientes FTP
- Servidores FTP
- El Protocolo FTP

- wu-ftpd
- ProFTPD

***Clientes FTP***

El usuario final accesa los servidores FTP desde un programa cliente. El cliente FTP más usado es el programa que se ejecuta desde la línea de comandos. Este inicia una conexión con un servidor, luego usted escribe comandos para recibir información y entonces da inicio a la transferencia de los archivos. El cliente traduce los comandos que usted escribe a comandos del protocolo FTP y se los envia al servidor. Este entonces interpreta las respuestas del servidor y se los presenta al usuario.

Otro cliente muy utilizdo en GNU/Linux es el cliente de FTP del navegador Mozilla. Aunque su uso principal es navegar en páginas de servidores HTTP, pero Mozilla también tiene pacidad de interactuar con servidores FTP. Si dirige al browser hacia una dirección URL del tipo ftp (e.j., ftp.codigolibre.org) o das click sobre un enlace a una URL de un FTP, Mozilla enviarea los comandos apropiados al servidor FTP y desplegará el listado de los directorios o descargará el archivo. Fijese de que manera podemos incluir el nombre de usuario y contraseña dentro del URL, algo similar a esto:

**ftp://nombreusuario:contraseña@ftp.redhat.com/**

Tome en cuenta que su contraseña será transferida en texto plano, asi es que tenga cuidado al usar esta manera de conectarse a un servidor.

Muchas distribuciones proveen clientes FTP que proveen interfaces gráficas y que son muy intuitivos, entre ellos se encuentran: gftp y NcFTP. Ellos proveen mecanismos de almacenar bookmarks a sitios muy visitados por el usuario y barras de progresos de las descarga, entre otras cosas para asistir la navegación. Estos también simplifican los comandos estándares de los clientes de línea de comandos y agregan características como son: completar nombres, descargar múltiples archivos, descarga en segundo planno, auto resumir descargar si falla, entre otras. Si usted deberá continuamente usar un cliente FTP, entonces lo más recomendable es usar uno de estos clientes.

***Servidores FTP***

Los servidores FTP permiten que otros sistemas remotos accesen a una porción de sus sistema de archivos. El FTP proveela misma funcionalidad del HTTP pero más orientada a la transferencia y no a la presentación de los archivos. El FTP es mucho mejor para la transferencia de archivos binarios y es muy frecuentemente usado para la descarga de software.

Un servidor FTP permite que solo usuarios autorizados usando un mecanismo estándar de nombre/contraseña puedan iniciar una sesión, o puede ser configurado para permitir el acceso a usuarios anonimos (guest users). Lo de usuario anonimo son referidos como anonymous FTP server y es la manera primordial de la transferencia de software Libre. El acceso a FTP Anonimo agrega una problematica de seguridad. Por ejemplo, si deseamos que los usuarios no puedan accesar archivos del sistema de archivos como es /etc/passwd. En la mayoría de los casos, usted tampo deseará que ellos puedan subir o escribir archivos al servidor ya que usted no controla quien se conecta por la cuenta anonimo.

Los servidores FTP muy a menudo utilizan la llamada del sistema chroot para mejorar los niveles de seguridad. La llamada chrootcambia el directorio raíz de un proceso. Después de la llamada chroot, los procesos no pueen aaceder ningún archivo fuera del subdirectorio que fué especificado en la llamada. El proceso nunca puede deshacer la llamada chroot y los procesos hijos heredarán el ambiente chrooted. El uso del chroot en los servicios FTP le permiten asegurarse que los usuarios anonimos sólo tendrán acceso limitado a un subconjunto de directorios en su sistema.

***El Protocolo FTP***

El protocolo FTP fué descrito por primera vez en el RFC 172 en el año 1971, aún antes de la implementación actual del TCP/IP fuese desarrollada. El protocolo actual está descrita en el RFC 959. Varias clarificaciones y actualizaciones se le han efectuado, incluyendo RFCs 1123, 1579, 2228, 2389, y el 2428.

Por RFC 854 se le asignan dos canales al FTP. Un canal es para los comandos y respuestas, y el otro es para la data y los archivos.

Una conexión FTP realmente consiste en dos conexiones, una para los comandos y otra para la data. La conexión de control se mantiene siempre abierta, pero la conexión de data sólo se abre cuando es requerida. La conexión de data del lado del servidor se encuentra en el puerto 20 y la de control se encuentra en el puerto 21. En realidad la conexión de data es abierta por el servidor, no el cliente. El cliente se enlaza con un puerto efímero y envia ese número al servidor. El servidor luego toma el puerto 20 de su lado y establece la conexión entre ese puerto y el puerto efímero del lado del cliente. La data del archivo entonces será pasada por esta conexión, con el fin de archivo definido por el que envia la data cuando ciuerra la conexión.

Básicamente, la manera que un FTP funciona es que el cliente inicia una conexión con un servidor. Este luego le envia comandos sobre esa conexión para recibir información y se inicia la transferencia de data. Por cada comando enviado, el servidor responde con resultados númericos además de una descripción en texto como resultado o cualquier error. La transferencia de data ocurre vía una segunda conexión. Al momento que una transferencia de data va a ocurrir, el clieente y el servidor negocian para establecer la conexión de data.

Existen dos modos deestablecer una conexión de data. En el estándar modo PORT, el cliente escucha en un puerto TCP y el servidor se conecta a el. Pero esto puede causar problemas con los firewalls, ya que las mayoría de cortafuegos no permiten conexiones entrantes. Asi que las nuevas versiones del protocolo FTP implementan modo pasivo. En este modo, el servidor escucha en un puerto por separado y el cliente efectuá una segunda conexión al servidor. La única diferencia entre los dos modos es si el cliente o el servidor inician el canal de data.

El FTP es un protocolo basado en comandos en que comandos ASCII son pasados entre el cliente y el servidor. Cuando un cliente FTP hace una conexión, este envía un comando ASCII al servidor. Estos comandos son quizas un poco primitivos y por lo general son mapeados a una sintaxis un poco más amistosa por la interfaz del cliente FTP. El servidor responde con un código numérico de tres dígitos y un mensaje opcional para indicar éxito, error, información de ayuda o la necesidad de más información.

Los que a continuación les presentamos son algunos de los comandos que son usados en el protocolo FTP:

| | |
|---|---|
| **USER** | Específica el nombre del usuario del login. |
| **PASS** | Específica el password/contraseña del login. |
| **CWD** | Cambia el directorio de trabajo. |
| **LIST** | Lista los archivos en el directorio. |
| **TYPE** | Específica el modo de transferencia de binario o texto. |
| **RETR** | Descarga (Retrives) un archivo. |
| **STOR** | Almacena (sube) un archivo. |
| **REST** | Reinicia una transferencia incompleta. |
| **DELE** | Elimina (Deletes) un archivo. |
| **QUIT** | Cierra la conexión. |

Recuerde que por lo general no tendrá la necesidad de utilizar estos comandos ya que el cliente le presentará un conjunto de comandos diferentes para efectuar las mismas acciones. Pero, si usted se conecta por telnet al servidor FTP en el puerto 21, usted puede usar estos comandos para ingresar al sistema y obtener cier-

ta información. Usted puede efectuar comandos raw y ver las respuestas raw, con los códigos de error y mensajes de texto. Usted puede ver un ejemplo de esta canal de comunicación raw en la sección anterior de telnet. Usted no podrá descargar archivos usando este m´todo porque el protocolo FTPrequiere el establecimiento de una segunda conexión para la transferencia de archivos.

**wu-ftpd**

Uno de los servidores FTP más usados es el wu-ftpd. Está disponible desde su sitio FTP propió, ftp.wu-ftpd.org. La mayoría de distribuciones de GNU/Linux vienen con wu-ftpd pre-instalado y a menudo es instalado con el nombre in.ftpd.

El servidor wu-ftpd provee muchas extensiones sobre el antiguo servidor basado en BSD FTP. Este provee capacidad de logging/diario, compresión dinámica (on-the-fly) y la capacidad de archivar (tar archiving), permisos por directorios de subir archivos, mensajes de login y directorio, limites de login basado en la clasificación de usuarios y hosting virtual.

La configuración del wu-ftpd se lleva acabo en varios archivos. Todos se encuentran en el directorio /etc por defecto. El archivo de configuración más importante es ftpaccess. Este define clases de usuarios y cualquier limitación colocada en esas clases. Algunos de los campos que pueden ser configurados en el archivos ftpaccess son:

- Dirección e-mail del Administrador
- Clases de Usuarios
- Limitaciones en el número de conexiones
- El archivo del mensaje de vienvenida
- Compresión y tar para todos los usuarios
- Diario/Log de las Transferencias
- Restricciones en las transferencias de archivos
- Restricciones en los comandos para los usuarios anonimos
- Habilitar y deshabilitar contraseña para los usuarios anonimos

Para habilitar el uso del archivo ftpaccess, deberá iniciar el daemon con la opción -a.

El primer pareametro que quizás desee establecer es las clases de los usuarios. Las clases de los usuarios dividen a los usuarios que se conectan al servidor para que asi le pueda asignar diferente ancho de banda y limitar su número de ingresos al sistema. Puede efectuar algo como este ejemplo a continuación:

```
class  local   real.anonymous   192.168.12.* * .codigolibre.org
class   remote  anonymous  *
limit  remote  10 Any /etc/ftp/demasiado.txt
```

El parámetro limit establece cuantos usuarios de cada clase pueden logearse en un espacio de tiempo y un mensaje a desplegarle si se le deniega el acceso. Usted puede permitir que suban archivos a directorios especificos utilizando los parámetros de subida. En este, usted puede especificar el directorio raíz, el directorio desde el punto de vista del usuario, si se puede subir archivos a este directorio, el dueño y el grupo que son los propietarios de los archivos y los permisos que deben recibir los archicos:

```
upload class=local   /home/ftp  /incoming  yes  owner  group  0644
```

El control sobre la compresión on-the-fly y archivar es administrada por el archivo ftpconversions. De hecho, en el usted específica la extensión del archivo que existe, la extensión que el usuario está requiriendo y el comando para convertir entre los dos, acompañado de algunas opciones. Para habilitar las características de tar y de comprimir, necesitará especificar las opciones de tar y compress en el archivo ftpaccess.

El archivo ftpusers específica los usuarios que existen en el archivo /etc/passwd que no son permitidos usar el servicio FTP. De cualquier otra forma, cualquier otro usuario en el archivo /etc/passwd está permitido usar el servicio de FTP siempre y cuando el shell especificado en el archivo passwd está listado en /etc/shells. Aquí le presentamos una parte de un archivo ftpusers:

news
nobody
root

Podemos ejecutar el wu-ftpd como un servidor stand-alone si le especificamos la opción -S. Al ejecutarlo stand-alone, usted puede especificarle el puerto en el que escuche con la opción -p. Por lo general, usted puede ejecutar wu-ftpd desde el inetd, con una línea parecida a la que sigue en el archivo inetd.conf:

**ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -a -l**

No se olvide de enviar una señal HUP al inetd si usted efecuá cualquier cambio a su archivo de configuración inetd.conf. Fijese que usamos unwrapper TCP sólo para ser consistente; la opción -a habilita el control de acceso basado en host usando el archivo ftpaccess. La opción -l nos indica que deseamos habilitar escritura al log.

## ProFTPD

Un popular servidor FTP de licencia GPL es el ProFTPD, disponible en su sitio web para descarga http://www.proftpd.net. Escrito desde cero con la intensión y meta de ser seguro y con más características que cualquier otro servidor FTP. Este usa un archivo central de configuración, con un grupo de directivas que deben ser muy familiar a todo administrador que está familiarizado con el Servidor Web Apache. El ProFTPD soporta logging, hosting virtual, autentificación PAM, integración SQL, integración con LDAP, mensajes de login y directorios entre muchas otras características.

La configuración .ftpaccess es similar a la que encontramos en el Apache en el archivo .htaccess. Estos archivos contienen los nombres de usuarios y su contraseña encriptada y son usados para restringir acceso a los usuarios a los directorios y archivos.

El Servidor ProFTPD puede ser ejecutado como un servicio stand-alone o desde el super daemon inetd. Para ejecutarlo desde el inetd, elimine cualquier otra línea que se refiera a otro FTP y reemplacela con la siguiente:

**ftp stream tcp nowait root /usr/sbin/tcpd in.proftpd**

Recuerde que deberá enviarle la señal al inetd de leer nuevamente si archivos de configuración:

**killall -HUP inetd**

### *Configuración*

Uno de los componentes más útiles del ProFTPD es la directiva AllowFilter. AllowFilter puede establecer una expresión regular que debe ser igualada para cualquier comando que sea enviado al ProFTPD. Esto le permite al administrador controlar con precisión cuales caracteres puede ser enviado al ProFTPD. El AllowFilter depende en igualar el argumento a una expresión regular, asi que el administrador debe ser muy cuidaso al crear la expresión. Si un comando falla la prueba de la expresión, el sistema retornará el mensaje de error “Forbidden Command” a la persona intgresada al sistema. Estos filtros son colocados en el archivo de configuración del servidor. Un ejemplo de una expresión de permitir (allow) que sólo aceptará caracteres alfanumericos y espacios es:

**AllowFilter “.*/[a-zA-Z0-9]+$”**

Esta entrada establece una expresión que busca caracteres que igualan cualquier de estos caracteres ., *, /, +, $, o cualquier letra mayuscula o minusculas y cualquier número.

Similar a la directiva AllowFilter, la directiva DenyFilter utiliza una expresión regular a igualar. La diferencia es que en el caso de DenyFilter el coamndo debe NO igualar la expresión. Si el comando enviado al servidor iguala la expresión de denegar, el usuario ingresado en el sistema recibirá el mensaje de error "Forbidden Command" ("Comando no Permitido"). Estos filtros también son establecidos en el archivo de configuración. Aquí le presentamos un ejemplo:

```
DenyFilter  "%?"
```

Este denegará cualquier comando enviado al servidor que contengan uno de los caracteres % o ?. Otra directiva es CommandBufferSize, la cual permite al administrador controlar el tamaño, een longitud de caracteres, que el servidor aceptará.

En resumen, el ProFTPD es un servidor FTP poderoso, que está disponible para GNU/Linux y todos los sistemas operativos Tipo-UNiX. Bajo constante desarrollo, con nuevas características y mejoras siempre disponible. Si un sistema esta ejecutando una versión que no es la actual, debe ser actualizada, porque como digimos ProFTPD está siempre mejorando. En particular, cualquier versión anterior a la 1.2.9 tiene cierto riesgos de seguridad que en las versiones superiores han sido todas corregidas. Todo sistema por seguro que sea tiene sus fallas asi es que porque ProFTPD es muy seguro no se descuide. Dos mejoras aún por materializarse son que no soporta contraseñas encriptadas y ID seguras, si su compañía requiere niveles de seguridad más estrictos, les acosajamos usar el VsFTPD. El cual si incluye las carecterísticas faltante de ProFTPD.

El servidor ProFTPD almacena su configuración en el archivo /etc/proftpd.conf. Aquí le presentamos un configuración de ejemplo. Discutiremos sólo los pareametros más comunes.

```
ServerName              "FTP de CodigoLibre"
ServerType              inetd
DefaultServer           on
Port                    21
MaxInstances            30
User                    nobody
Group                   nobody
Umask                   022
<Directory /*>
    AllowOverwrite      no
</Directory>
<Anonymous  /home/ftp>
User                    ftp
Group                   ftp
UserAlias               anonymous ftp
MaxClients              10
 <Limit WRITE>
    DenyAll
 </Limit>
</Anonymous>
```

El parámetro ServerName específica el nombre a ser desplegado cuando un cliente se conecta al servidor. Es desplegado antes del prompt de nombre de usuario contraseña. El ServerType específica si el servidor será ejecutado desde inetd o si será en modo de stand-alone. La directiva MaxInstances puede ser usada para limitar el número de conexiones que un servidor puede aceptar. Los parámetros de User y Group dan el nombre del usuario bajo el cual el servidor se ejcuta. Una vez el servidor se ha establecido y no necesita ya los permisos de root, este cambiará su identidad al usuario y al grupo aquí señalado. La sección Directory establece los parámetros que deben ser aplicados a un directorio en particular. En nuestro ejemplo, queremos que los usuarios puedan sobreescribir archivos y permitiremos eso en cualquier subdirectorio del directorio root.

La sección que contiene la palabra clave Anonymous establece el acceso de los usuarios anonimos, ejecutando como el usuario de nombre ftp. Hasta 10 usuarios anonimos son permitidos a la vez, usando o el nombre "anonymous" o el nombre "ftp". La directiva Limit niega el permiso a los usuarios anonymous de subir o escribir archivos al servidor.

### Ejercicio 7-3: Configurar los Servicios de FTP Usando ProFTPD

Las soluciones a este ejercicio se incluyen en su contenido.

1. Instale el paquete de ProFTPD. Si su distribución viene ya con el incluido en una versión pre-empaquetada use esa. Sino descargue el paquete desde el sitio web de ProFTPD http://www.proftpd.net, descomprimale, compilelo e instalelo.

2. Debe crear un directorio para el acceso anónimo:
   **# mkdir /home/ftp**

3. Edite el archivo /etc/proftpd.conf. Primero configure el servidor para solo tener acceso de usuarios autorizados:

```
ServerName              "FTP de CodigoLibre"
ServerType              inetd
DefaultServer           on
Port                    21
MaxInstances            30
User                    nobody
Group                   nobody
Umask                   022
<Directory /*>
    AllowOverwrite      no
</Directory>
```

4. Edite el archivo /etc/inetd.conf. Elimine el simbolo de comentario de la línea que pertiene a cucalquier servicio de FTP actualmente. Agregue la siguiente línea:
   **ftp stream tcp nowait root /usr/sbin/tcpd in.proftd**

   Usamos wrappers TCP aquí, wlo cual no es estrictamente necesario ya que ProFTPD puede efectuar control de acceso a los hosts por si mismo, pero ers bueno ser consitente y usar wrappers TCP en todos los servicios.

5. Le enviamos una señal al inetd para que lea nuevamente su archivo de configuración:
   **# killall -HUP inetd**

6. Pruebe el servidor ejecutando un cliente FTP. Es mejor provar desde un equipo diferente, siempre y cuando sea posible. Será necesario suplir un nombre de usuario y contraseña válidos en el sistema.
7. Habilite el acceso anónimo agregando las siguientes líneas al final de su archivo /etc/proftpd.conf:

```
<Anonymous /home/ftp>
User                    ftp
Group                   ftp
UserAlias               anonymous ftp
MaxClients              10
 <Limit WRITE>
    DenyAll
 </Limit>
</Anonymous>
```

8. Pruebe el servidor ejecutando un cliente FTP, desde un computador diferente si es posible. Ingrese (login) con el nombre de usuario “ftp”. Se le pedirá en el prompt entrar su dirección de correo en vez de una contraseña.

## SQUID

Si cada petición de un objeto Web requiriera una conexión al servidor actual efectuando el hosting y la recarga completa del objeto desde la data de origen, su navegación en el Internet fuese lentísima y frustratoria. Lo que hace falta es un método de capturar (caching) los objetos Web más frecuentes y recientes. Bueno este tipo de software ya existe y es conocido como caching proxy server (Servidores Proxy de Captura).

Un servidor proxy es un servidor transparente que sirve como mediador entre una red privada y el Internet. Los usuarios navegan vía el servidor proxy para accesar la información desde el Internet. Si un usuario hace una petición de una página Web en el Internet desde una red protegida, la petición irá al servidor proxy antes de ser pasada al servidor real. El servidor proxy entonces revisa para ver si la página requerida está en su cache. Si la página está en el cache del proxy, el servidor proxy le envia esta página directamente. hay funciones adicionales que efectuá un servidor proxy que serán explicadas en las siguientes secciones.

Hay diferentes métodos de configurar un servidor proxy. Una de las configuraciones más popular de configurar un servidor proxy es en la que tanto como las conexiones entre el Internet y la red protegida son atraves del servidor proxy. En esta instancia, un servidor proxy es nada más que un programa que reside en el firewall (corta fuego) y puede ver ambos lado de la interfaz: la intranet y el Internet. Aunque muy a menudo el firewall y el servidor proxy por lo general aparecen juntos, no son la misma cosa. El firewall filtrador de paquetes opera en la capa de Red del modelo OSI, mientras que los servidores proxy trabajan en la capa de Aplicación.

Un firewall filtrador de paquetes puede bloquear varios protocolos y direcciones IP de entrar a nuestra red local. Este también puede controlar los protocolos. Este también puede controlar los protocolos que los usuarios son permitidos tener acceso al conectarse al Internet.

Los proxies pueden ser clasificados especificos a su aplicación. Algunos ejemplos de servidores proxy son un proxy HTTP para las páginas Web, un servidor proxy FTP,un proxy SMTP/POP para los e-mails, un proxy NNTP (Network News Transfer Protocol) para los servidores de noticias, entre otros.

GNU/Linux incluye un proxy cache de páginas Web muy popular llamado Squid. En esta sección, cubriremos los siguientes tópicos:

- Configurar Squid
- Iniciar y Detener Squid en un Servidor GNU/Linux
- Características de Seguridad de los Servidores Proxy
- Función de Caching
- ACL (Access Control List/Listas de Control de Acceso)

***Configurar Squid***

Squid Puede ser configurado dependiendo de las necesidades de un site. El archivo de configuración de squid se encuentra en el directorio /etc/squid/. Algunas de las opciones comúnmente modificadas por el administrador en el arcivo de configuración se muestra aquí en lo adelante:

**cache_dir Type Dirname Mbytes Level1 Level2**

La función de esta opción es la de especificar el nombre del directorio en el cual se almacenará el cache.