UNIVERSIDAD DEL AZUAY

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas

Monografía Previa La obtención
Del título De Ingeniero De Sistemas

Tema: Emulación de Routers mediante Pc's genéricos y utilización del Sistema Operativo Linux

Autores:

Fernando Aguilar O.
Fabricio Contento A.

Cuenca, Agosto 26 de 2004

DEDICATORIA

Por su labor diaria, fuente inagotable de sacrificio y amor, que son mi inspiración para siempre seguir adelante.

Por enseñarme a valorar lo que significa sacrificarse por un ser querido.

Este trabajo esta dedicado a mis padres por su amor y apoyo constante, quienes en todo momento creyeron en mí.

Fabricio Contento Astudillo.

La culminación de este paso muy importante en mi vida pero que es el inicio de muchos sueños para los cuales he trabajado y lo seguiré haciendo el resto de ella, se lo dedicó al ser más importante de mi vida que es mi Señora Madre, gracias por todo lo que me haz enseñado, la honradez, la humildad, la amistad en fin todas esas virtudes que posees y que he tratado de hacerlas parte de mi vida.

Me faltan palabras para decirte lo que siento pero este triunfo aunque pequeño espero que en algo recompense todos tus sacrificios.

"Todos los triunfos nacen cuando nos atrevemos a comenzar"

Fernando Aguilar Ochoa

AGRADECIMIENTO

El presente trabajo representa la culminación de todo el esfuerzo de superación que lo emprendimos el momento de iniciar nuestra carrera en la Universidad

del Azuay, durante todo el proceso de aprendizaje adquirimos muchas cosas valiosas que van más allá de la mera educación técnica. Pudimos crecer como personas valorando el sacrifico que han hecho nuestros padres.

En primer lugar agradezco a mis padres, por su apoyo incondicional durante toda mi vida, sin su esfuerzo y sin su cariño no hubiese podido llegar a alcanzar una de mis metas que se ha cumplido.

A mis hermanos, primos, amigos y seres queridos, que de una y otra manera han contribuido con su cariño, apoyo y amistad, quienes me han demostrado que personas valiosas y desinteresadas aún existen.

A mis profesores, quienes durante toda mi vida universitaria supieron impartir en mí no solo el conocimiento científico, sino la semilla de siempre estar aprendiendo; que de lo más sencillo de la vida se puede obtener algo valioso y el conocimiento adquirido es para transmitirlo, de que sirve el esfuerzo de aprender si al final morirá con nosotros.

Fabricio Contento Astudillo.

Mi agradecimiento es para todos mis familiares y amigos, quienes siempre estuvieron allí cuando necesite apoyo, en especial a mis tías Jenny y Alicia las mismas que fueron una Segunda Madre para mi y siempre estuvieron pendientes de mis acciones.

A mis profesores porque siempre trataron de enseñarnos el lado humano de las cosas conjuntamente con el conocimiento científico, como también ha cultivar ese precioso fruto que es el conocimiento y lo principal de algo que se cultiva compartirlo con los demás.

Estoy tranquilo porque se que en este mundo aun existen personas valiosas como mi gran amigo Fabricio que fue uno de los pilares fundamentales en esta Odisea que emprendimos hace 5 años y que ahora estamos culminando, gracias por soportar ha este ser humano con múltiples defectos.

"Quien no es capaz de construir castillos en el aire, no es capaz de habitar en ellos"
Fernando Aguilar Ochoa

INTRODUCCION

El presente trabajo expone una solución al alto costo que se tiene en la construcción de redes para empresas principalmente PYMES.

Dando lugar a la creación de Routers por medio de PC's lo que le equivale a la empresa un ahorro considerable y una funcionalidad elevada, utilizando computadores genéricos, lo que le permite trabajar eficientemente y tener escalabilidad y flexibilidad en la ampliación de sus redes.

Generalmente los ruteadores utilizados son dispositivos de propósito específico y de algunas marcas determinadas, por lo tanto teniendo costos elevados en la adquisición de los mismos y en su mantenimiento.

Hay otras alternativas como el uso de software libre para llevar a cabo estas tareas, una de las alternativas es el uso del sistema operativo de código abierto Linux, que por estabilidad y bajo costo, se tiene que conocer muy bien el sistema operativo Linux para la administración del sistema.

La utilización de software libre nos permite tener las mismas funcionalidades que los productos comerciales, nos da la opción de un firewall integrado, costos reducidos, desarrollo continuo y estandarizado, y además respuesta inmediata a bugs.

Se tiene que acotar que el uso de LINUX representa que se va ha llevar las funciones de ruteo y del sistema operativo conjuntamente, y que hay tener muchas consideraciones en el aspecto de configuración y mantenimiento del sistema.

Presentamos el Sistema Operativo LINUX como la solución a los problemas de costos y prestaciones en la instalación de redes medianas, y de baja carga. Se prodría diseñar arquitecturas más complejas para las grandes empresas teniendo un costo muy bajo su implementación pero se necesitaría personas con un conocimiento profundo sobre la arquitectura de LINUX.

INTRODUCCION

1.1 Introducción

El protocolo TCP/IP permite a diferentes computadoras con diversos sistemas operativos comunicarse entre si. Esto comenzo en los años 60 como un proyecto de investigación financiado por el gobierno de EEUU dentro del paquete switching networks que tenia, en los 1990s, llego hacer el mas usado entre las redes de computadoras. Este es un sistema abierto donde los protocolos y sus implementaciones casi no son modificados. Lo que formó la base para que sea llamado el *worldwide Internet,* o el *Internet,* una wide area network (WAN).

1.2 Capas

Normalmente los protocolos de red son desarrollados en capas, con cada capa responsable para una difernte faceta de las comunicaciones. Un *protocolo, como* TCP/IP, es la combinación de diferentes protocolos en varias capas. TCP/IP se caracteriza por ser un sistema de 4 capas, como se muestra en la Figura 1.1.

Application	Telnet, FTP, e-mail, etc.
Transport	TCP, UDP
Network	IP, ICMP, IGMP
Link	device driver and interface card

Figura 1.1 Los cuator layers del protocolo TCP/IP.

En la capa de red de TCP/IP, IP, provee un servicio no confiable. Eso es, realiza el mejor trabajo en el movimiento de paquetes hacia su destino, pero no provee garantias. TCP, por el otro lado, provee un transporte confiable de capas usando el servicio de IP para proveer este servicio. TCP ejecuta el timeout y retrasmicion, envia y recibe end-to-end acknowledgments. La capa de red y la capa de transporte tienen distintas responsabilidadaes.

Un router, por definición, tiene dos o mas capas de interface de red (desde que se conecta en dos o mas redes). Todo sistema con multiple interface es llamado *multihomed.* Un host puede también ser multihomed pero este especifica los forwards packets desde una interfase a otra, por lo que no es un router. También, los routers no necesitan estar en cajas especiales de hardware que solo mueven paquetes al rededor de una red. La mayoria de las implementaciones TCP/IP permiten a un host multihomned actuar como un router. Para lo cual el host necesita estar especificamente configurado para que esto ocurra. En este caso podemos llamar al sistema como un host (cuando una aplicación como FTP o TELNET esta siendo usada) o un router (cuando sus forwarding packets estan de una red a otra).

Una de las metas de una Internet es esconder todos los detalles de las capas fisicas de la red desde las aplicaciones. Aunque esto no es obvio desde nuestras dos redes de trabajo, las capas de aplicación no pueden cuidar (y no cuidan) que un host este sobre una Ethernet, el otro sobre una token ring, con un router entre ellos. Esto podría ser entre 20 routers, con tipos adicionales de interconecciones fisicas, y las aplicaciones correrian al mismo tiempo. Este ocultamiento de los detalles es lo que crea el concepto una Internet poderosa y útil.

Otra vía para conectar redes es con un *bridge*.Este conecta redes mediante la capa de enlace, mientras los routers conectan las redes mediante la capa de red. Los bridges crean multiples LAN's que aparecen sobre las capas como una única LAN. TCP/IP internet tiende a ser construido mediante routers en lugar de bridges.

1.3 Capa TCP/IP

TCP y DP son ls protocolos predominantes en la capa de transporte. Ambos utilizan IP sobre la red. TCP provee una capa de transporte confiable, aunque el servicio de IP no lo sea.

UDP envia y recibe datagramas para aplicaciones. Un datagrama es una unidad de información que viaja desde un trasmisor hacia un receptor.UDP no es orientado a conexión. Lo que no garantiza que un datagrama llegue a su destino.

IP es el protocolo principal en la capa de red.Es utilizado por TCP y UDP. Toda trama de TCP y UDP que logra ser transferida alrededor de una reed va através de una capa IP hacia los sistemas finales y los routers intermedios.

ICMP es un adjunto para IP. Estye es usado por IP para el intercambio de mensajes de error y otra información con priorodad con otra capa IP en un host o router.

IGMP es el Internet Group Management Protocol. Este es usado como multicasting: enviando un datafgrama UDP a multiple hosts.

ARP (Address Resolution Protocol) and RARP (Reverse Address Resolution Protocol) son protocolos especializados usados solo con ciertos tipos de interfaces de red (como Ethernet y token ring) para resolover las direcciones entrela capa IP y las interfaces de red.

1.4 Direcciones en Internet

Toda interface sobre una internet debe tener una única dirección (también llamada *IP address).*Esta direcciones son números de 32 bits. En lugar de usar un espacio de direcciones flat tal como 1, 2, 3, hay una estructura para las direcciones de internet.

Estas direcciones de 32 bits son escritas como 4 numeros decimales, uno para cada byte de la dirección. Esto es llamado numeración punto-decimal. Por ejemplo, una dirección de clase B es 140.252.13.33.

La via más fácil para difernciar entre las distiontas clases de direcciones es observer el primer número deuna dirección de punto-decimal. La figura muestra las diferentes clases con el primer número realzado.

Clase	Rango
A	**0**.0.0.0 to **127**.255.255.255
B	**128**.0.0.0 to **191**.255.255.255
C	**192**.0.0.0 to **223**.255.255.255
D	**224**.0.0.0 to **239**.255.255.255
E	**240**.0.0.0 to **247**.255.255.255

Clases de direcciones IP con sus diferentes rangos

Este es un valor reiterativo que un host multihomed tendrá multiples direcciones IP: una por interface.

Cada interface sobre una internet de tener una única dirección IP, existe una autoridad central que designa esta direcciones para las redes conectadas al Internet. Esta autoridad es *Internet Network Information Center,* llamado InterNIC. El InterNIC asigna solo identificaciones de red. La asignación de los ID's de host esta sobre el administrador de sistema.

Hay tres tipos de direcciones IP: *unicast* (destinado para un solo host), *broadcast* (destinado para todos los host dentro de una red), y *multicast* (destinados para un conjunto de hosts que pertenecen a un grupo multicast).

1.5 El Domain Name System

Aunque la interface de red sobre un host, y por tanto el host por el mismo, son conocidos por las direcciones IP, las personas trabajan mejor usando el nombre del host. En el mundo TCP/IP el *Domain Name System* (DNS) es una base de datos distribuida que prove el mapeo entre las direcciones IP y los nombres de host.

Por ahora debemos estar concientes que ninguna aplicación puede llamar a una función de una librería standard para ver sobre la dirección IP (o direcciones) correspondiente a un hjostname dado. Similarmente una function es proveida para hacer lo contrario dado una dirección IP, viendo sobre el correspondiente hostname.

La mayoria de las aplicaciones que toman a un hostname como un argumento también toman una dirección IP. Cuando usamos el Telnet,por ejemplo, una vez especificamos un hostname y en otra ocasión especificamos una dirección IP.

1.6 Encapsulamiento

Cuando una aplicación envía un dato utilizando TCP, el dato es enviado bajo el stack del protocolo, atravesando cada capa, hasta que este es enviado como una cadena de bits a través de la network. Cada capa adiciona información a los datos sobre sus headers (y algunos adicionan un contenedor de información) para los datos que recibe. La unidad de datos que TCP envia sobre IP es llamado un segmento *TCP.* La unidad de datos que IP envia a la interface de red es llamado datagrama IP*.* La cadena de bits que viajan atravéz de la red es llamado *frame.*

Los numeros en el fondo de los headers y contenedores del frame de la Ethernet son los tamaños tipícos de los headers en bytes.

Una propiedad física del frame de una Ethernet es que el tamaño de sus datos deben estar entre 46 y 1500 bytes.

Esto es similar en UDP salvo que su unidad de información es llamada datagrama UDP*,* y el tamaño del UDP header es 8 bytes.

Recalcando que TCP, UDP, ICMP, y IGMP todos envian datos sobre IP. IP debe adicionar algún tipo de identificador para el IP header que este genera,para indicar la capa a la que cada dato pertenece. IP maneja esto por almacenamiento en un valor de 8 bits en este header llamado *protocol* field. Un valor de 1 es para ICMP, 2 para IGMP, 6 indica TCP, y 17 es UDP.

Igualmente diferentes aplicaciones pueden estar usando TCP o UDP al mismo tiempo. Los protocolos de la capa de transporte almacenan un identificador en los headers que ellos generan para identificar la aplicación. TCP y UDP usan numeros de puertos de 16-bit *port numbers* para identificar aplicaciones. TCP y UDP almacenan el número de puerto origen y destino en sus respectivos headers.

1.7 Demultiplexado

Cuando un frame Ethernet es recibido en el host destino este comienca ha ir hacia arriba en el stack del protocolo y todos los headers son removidos por el protocolo apropiado. Cada protocolo busca ciertos identificadores en este header para determiner que protocolo en la capa superior recibe los datos. Esto es llamado *demultiplexing.*

1.8 Modelo Cliente - Servidor

Varias aplicaciones de red estan escritas asumiendo en un lado al cliente y en el otro el servidor. El propósito de la aplicación es que el servidor provea algún servicio definido para los clientes.

Se puede categorizer los servidores en dos clases: interactivo o concurrente. Un server interactico itera mediante los siguientes pasos:

1. Espera por una solicitud del cliente.

2. Procesa la solicitud del cliente.

3. Envia la respuesta al cliente que envió la solicitud.

4. Regresa al paso 1.

El problema con un server iterativo es cuando esta en el paso 2. Durante este tiempo no es atendido otro cliente.Un server concurrente, en el otro casa, ejecuta los siguientes pasos.

l. Espera por la solicitud de un cliente.

2. Empieza un Nuevo server para atender la solicitud del cliente. Esto lo hace creando un nuevo proceso, tarea, o hilo, dependiendo sobre que soporta el sistema operativo base.

El Nuevo server maneja las solicitudes de entrada del cliente. Cuando termine de responder a las solicitudes del cliente este termna.

3. Regresa al paso l.

La ventaja de sistema concurrente es que el server solo genera nuevos servers para atender las solicitudes de los clientes. Cada cliente tiene, en escencia, su propio server. Asumiendo que el sistema operativo soporta multiprogramación, son atendidos varios clients en forma concurrente.

Como una regal general, los servers TCP son concurrentes, y los servers UDP son iterativos, con algunas excepciones.

1.9 Numeros de Puerto

Los servicios son conocidos por los numeros de puertos. Por ejemplo, toda implementación TCP/IP que prove un servicio FTP lo hace por el Puerto 21 de TCP. Todo servicio TELNET esta sobre el puerto 23 de TCP. Toda implementación de TFTP (the Trivial File Transfer Protocol) esta sobre el puerto 69 de UDP. Estos servicios que pueden ser proveidos por alguna implementación TCP/IP tiene bien conocidos los numeros de puertos que estan entre 1 y 1023.

Un cliente usualmente no se percado sobre el número de Puerto que hasta usando para su fin. Los port numbers del cliente son llamodos puertos efímeros. Esto es porque un cliente tipicamente existe solo mientras esta corriendo el servicio que necesita, mientras que los servicios corren mientras el host esta levantado.

Muchas implementaciones TCP/IP permiten puertos efímeros entre 1024 y 5000. Los port numbers sobre el 5000 son utilizados por otros servicios.

Los Puertos Reservados estan en el rango de 1 a 1023, y son usados por algunas aplicaciones, como parte de la autentifiación entre el cliente y el servidor.

1.10 Proceso de Estandarización

Hay cuatro grupos responsables del Internet technology.

1. The *Internet Society* (ISOC)
2. The *Internet Architecture Board* (IAB)
3. The *Internet Engineering Task Force* (IETF)
4. The *Internet Research Task Force* (IRTF).

1.11 RFCs

Todos los standares oficiales en la comunidad de internet son publicados como un *Request for Comment,* o *RFC.*

Todos los RFCs estan habilitados y pueden obtenerse enviando un mail como se muestra:

To: rfc-info@OISI.EDU
Subject: getting rfcs

help: ways_to_get_rfcs

Retorna un listado detallado de la manera de obtener los RFC's.

Algunos RFC's importantes son:.

1. El *Assigned Numbers RFC* especifica los numeros y constantes que son usados en los protocolos de internet.Todos los port numbers conocidos estan listados aqui.
2. El *Internet Official Protocol Standards,* concurrentemente RFC 1600. Este RFC especifica el estado de estandarización de los varios protocolos de internet. Cada protocolo tiene un estado de estandarización de los siguientes: standard, draft standard, proposed standard, experimental, informational, o historic. Adicionalmente cada protocolo tiene un nivel de requerimiento: required, recommended, elective, limited use, o not recommended.
3. El Host *Requirements RFCs,* 1122 y 1123. RFC 1122 maneja la capa de enlace, capa de red y capa de transporte, mientras el RFC 1123 maneja la capa de aplicación.
4. El *Router Requirements RFC.* RFC 1009, es similar a los RFC's de los requerimientos de hosts, pero especifica los unicos requerimientos de los routers.

1.12 Standard, Simple Services

Son pocos standares,simples servicios que permiten proveer todas las implementaciones.

Nombre	TCP port	UDP port	RFC	Descripción
echo	7	7	862	Servicio que retorno sea lo que sea que envoi el cliente.
discard	9	9	863	Server descarta sea lo que sea que envio el cliente.
daytime	13	13	867	Server retorna hora y fecha
chargen	19	19	864	TCP server envia una cadena continua de caracteres, hasta que la conexión es terminada por el cliente. UDP server envia un datagrama conteniendo un número randómico de caracteres cada vez que el cliente envia un datagrama.
time	37	37	868	Server retorna la hora como un número binario de 32-bit.

Figura 1.3 Standard, servicio simples proveidos por varias implementaciones.

Capa de Enlace

2.1 Introducción

El propósito de la capa de enlace en el conjunto de protocolos TCP/IP es enviar y recibir (1) datagramas IP para los módulos IP, (2) ARP requests and replies para el módulo ARP, y (3) RARP requests and replies para el módulo RARP. TCP/IP soprta diferentes capas de enlace, dependiendo del tipo de hardware de red que esta siendo usado: Ethernet, token ring, FDDI (Fiber Distributed Data Interface), RS-232 serial lines, etc.

2.2 Ethernet y Encapsulación IEEE 802

El término *Ethernet* generalmente se refiere a un Standard publicado en 1982 por Digital Equipment Corp., Intel Corp., y Xerox Corp. Esta es la forma predominante en LAN's usada con TCP/IP hoy en día. Esta usa un método de acceso llamado CSMA/CD y opera a 10 Mbits/sec usando direcciones de 48-bits.

En el mundo TCP/IP, la encapsulación de datagramas IP está definida en RFC 894 para Ethernets y en RFC 1042 para redes IEEE 802.

Los dos formatos de frame usan direcciones origen y destino de 48-bit (6-byte). A estos se los llama *hardware addresses* a lo largo del texto. Los mapas de protocolos ARP y RARP estan entre las direcciones IP de 32-bit y las direcciones de hardware de 48-bit.

Los próximos 2 bytes son diferentes en los dos formatos de frame. El campo *length* en 802 dice como van muchos bytes siguientes, pero no incluye el CRC al final. El campo *type* en Ethernet identifica el tipo de dato que sigue. En el frame de 802 el mismo campo *type* ocurre después en el header de SNAP (Sub-network Access Protocol). Afortunamente ninguno de los valores válidos en el campo *length* de 802 es el mismo que los valores del campo *type* en Ethernet, haciendolos a los dos formatos de frames distinguibles.

En el frame Ethernet los datos inmediatamente permiten el campo *type*, mientras en el formato del frame 802 permite 3 bytes de LLC en 802.2 LLC y 5 bytes de SNAP en 802.2. El *DSAP* (Destination Service Access Point) y el *SSAP* (Source Service Access Point) son seteados a 0xaa. El campo *Ctrl* es seteado a 3. Los próximos 3 bytes, el código *org* estan todos en 0. Revisando esto es el mismo campo *type de 2 bytes* que tenemos en el formato del frame de Ethernet. El campo *CRC* es un cyclic redundancy check (un checksum) que detecta errors en el resto del frame. (también llamado *FCS* o frame check sequence.)

Hay un tamaño minimo para frames en 802.3 y Ethernet. Este minimo require que la porción de datos sea menos det 38 bytes para 802.3 o 46 bytes para Ethernet.

2.3 Trailer Encapsulation

RFC 893 describe otra forma de encapsulación usada sobre Ethernets, llamada *trailer encapsulation.* "Los campos de tamaño variable en el comienzo de la porción de los datos del frame Ethernet (el IP header y el TCP header) fueron movidos al final (hacia la derecha antes del CRC). Esto permite que la porción de datos del frame sea mapeada hacia una hardware page, guardando una copia memory-to-memory cuando el dato es copiado en el kernel. El dato TCP que es de 512 bytes de tamaño es movido para solo ser manipulado por las páginas de tablas del kernel. Dos hosts negocian el uso de trailer encapsulation usando una extensión de ARP.

2.4 SLIP: Serial Line IP

SLIP permanence por una línea serial IP. Esta es una forma simple de encapsulación para datagramas IP sobre líneas seriales, y es específico. SLIP es popular para la conexión de sistemas caseros al Internet.

2.5 Compressed SLIP

Desde que las lineas SLIP son frecuentemente lentas (19200 bits/sec o menos) y usadas generalemte para tráfico interactivo (tal como Telnet y Rlogin, que usan TCP), esta tendiendo a que muchos paquetes pequeños TCP sean intercambiados a través de lineas SLIP. Para acarrear I byte de datos require un IP header de 20 bytes y un TCP header de 20 bytes, un overhead de 40 bytes. Reconociendo este performance, una nueva version de SLIP, llamada CSLIP (compressed SLIP), esta especificada en RFC 1144. CSLIP normalmente reduce los 40 bytes de header a 3 o 5 bytes. It maintains the state of up to 16 TCP connections on each end of the CSLIP link and knows that some of the fields in the two headers for a given connection normally don't change. Of the fields that do change, most change by a small positive amount. These smaller headers greatly improve the interactive response time.

2.6 PPP: Point-to-Point Protocol

PPP, Point-to-Point Protocol, corrige todas las deficiencias de SLIP. PPP consiste de tres componentes.

1. Una manera para encapsular datagramas IP sobre un enlace serial. PPP supports either un enlace asíncrono con 8 bits de datos y sin paridad o enlaces síncronos orientados.
2. Un *link control protocol* (LCP) para establecer, configurar, y testear la data-link connection. Esto permite que en cada final se pueda negociar varias opciones.
3. Una familia de *network control protocols* (NCPs) especifica para diferentes protocolos de capa de red. Los RFCs concurrentemente existen por IP, la capa de red OSI, DECnet, y AppleTalk. El IP NCP, por ejemplo, permite especificar en cada final si este puede ejecutar compresión de header, similar a CSLIP.

Desde que PPP, como SLIP, es frecuentemente usado atravéz de enlaces serials lentos, reduciendo el número de bytes por frame reduce la latencia por las aplicaciones interactivas. Usando el link control protocol, varias implementaciones negocian para omitir las direcciones constants y los campos de control y para reducir el tamaño del campo del protocolo de 2 bytes a 1 byte. Si comparamos el framing overhead en un frame PPP, versus los 2-byte de framing overhead en un frame SLIP, veremos que PPP addiciona tres bytes: I byte para el *protocol* field, y 2 bytes para el CRC. Adicionalmente, usando el IP network control protocol, varias imlementaciones negocian para usar Van Jacobson header compression (identico al CSLIP compression) para reducir el tamaño de los headers de IP y TCP.

En conclusión, PPP prove las siguientes ventajas sobre SLIP: (1) soporta multiples protocolos sobre una única línea serial, no solo datagramas IP, (2) Un CRC en cada frame, (3) negociación dinámica de las direcciones IP en cada punto(usando el IP network control protocol), (4) compresión de header TCP e IP similar a CSLIP, y (5) un link control protocol para las negociaones de varias opciones de data-link.

2.7 Loopback Interface

Varias implementaciones soportan un *loopback interface* que permite al cliente y al servidor sobre el mismo host comunicarse con ellos mismos usando TCP/IP. La red de clase A con ID 127 es reservado para el loopback interface. Por convención, varios sistemas asignan la dirección IP 127.0.0.1 para esta interface y le asignan el nombre localhost. Un datagrama IP enviado al loopback interface no debe aparecer en otro red.

2.8 MTU

Hay un límite en el tamaño para los frames del Ethernet encapsulation y 802.3 encapsulation. Esto limita el número de bytes de datos a 1500 y 1492, respectivamente. Esta característica de la capa de enlace es llamada *MTU,* esta es la máxima unidad de trasmisión.

Si IP tiene un datagrama para enviar, y el datagrama es más grande que la MTU de la capa de enlace, IP ejecuta fragmentación, rompiendo el datagrama en varios pedazos (fragmentos), haciendo que cada fragmento sea más pequeño que la MTU.

2.9 Path MTU

Cuando dos hosts sobre la misma red están comunicandose el uno con el otro, este es el MTU de la red. Pero cuando dos hosts están comunicándose a través de múltiples redes, cada enlace puede tener una diferente MTU. Los numeros importantes no son las MTUs de las dos redes para que los hosts se conecten, pero si la MTU más pequeña que será el tamaño de referencia para la conexión. Este es llamado *path MTU.*

El path MTU entre dos hosts cualesquiera necesita no ser constante. También, el routing no debe ser simétrico (la ruta desde A a B no debe ser el contrario a la ruta desde B hacia A), por lo tanto el path MTU necesita no ser el mismo en las dos direcciones.

IP: Internet Protocol

3.1 Introducción

El protocolo IP es el software que implementa el mecanismo de entrega de paquetes sin conexión y no confiable (técnica del mejor esfuerzo).

El protocolo IP cubre tres aspectos importantes:

1. Define la unidad básica para la transferencia de datos en una interred, especificando el formato exacto de un Datagrama IP.
2. Realiza las funciones de enrutamiento.
3. Define las reglas para que los Host y Routers procesen paquetes, los descarten o generen mensajes de error.

3.2 IP Header

Los Datagramas IP están formados por *Palabras* de 32 bits. Cada Datagrama tiene un mínimo (y tamaño más frecuente) de cinco palabras y un máximo de quince.

Ver	Hlen	TOS	Longitud Total	
Identificación			Flags	Desp. De Fragmento
TTL		Protocolo	Checksum	
Dirección IP de la Fuente				
Dirección IP del Destino				
Opciones IP (Opcional)				Relleno
DATOS				

Ver: Versión de IP que se emplea para construir el Datagrama. Se requiere para que quien lo reciba lo interprete correctamente. La actual versión IP es la 4.

Hlen: Tamaño de la cabecera en palabras.

TOS: Tipo de servicio. La gran mayoría de los Host y Routers ignoran este campo. Su estructura es:

Prioridad	D	T	R	Sin Uso

La prioridad (0 = Normal, 7 = Control de red) permite implementar algoritmos de control de congestión más eficientes. Los tipos D, T y R solicitan un tipo de transporte dado: D = Procesamiento con retardos cortos, T = Alto Desempeño y R = Alta confiabilidad. Nótese que estos bits son solo "sugerencias", no es obligatorio para la red cumplirlo.

Longitud Total: Mide en bytes la longitud de doto el Datagrama. Permite calcular el tamaño del campo de datos: Datos = Longitud Total – 4 * Hlen.

TTL: Tiempo de Vida del Datagrama, especifica el numero de segundos que se permite al Datagrama circular por la red antes de ser descartado.

Protocolo: Especifica que protocolo de alto nivel se empleó para construir el mensaje transportado en el campo datos de Datagrama IP. Algunos valores posibles son: 1 = ICMP, 6 = TCP, 17 = UDP, 88 = IGRP (Protocolo de Enrutamiento de Pasarela Interior de CISCO).

Checksum: Es un campo de 16 bits que se calcula haciendo el complemento a uno de cada palabra de 16 bits del encabezado, sumándolas y haciendo su complemento a uno. Esta suma hay que recalcularla en cada nodo intermedio debido a cambios en el TTL o por fragmentación.

Dirección IP de la Fuente:

Dirección IP del Destino:

Opciones IP: Existen hasta 40 bytes extra en la cabecera del Datagrama IP que pueden llevar una o más opciones. Su uso es bastante raro.

- Uso de Ruta Estricta (Camino Obligatorio)
- Ruta de Origen Desconectada (Nodos Obligatorios)
- Crear registro de Ruta
- Marcas de Tiempo
- Seguridad Básica del Departamento de Defensa
- Seguridad Extendida del Departamento de Defensa

3.3 IP Routing

Enrutar es el proceso de selección de un camino para el envío de paquetes. La computadora que hace esto es llamada Router.

En general se puede dividir el enrutamiento en ***Entrega Directa*** y ***Entrega Indirecta***. La Entrega Directa es la transmisión de un Datagrama de una maquina a otra dentro de la misma red física. La Entrega Indirecta ocurre cuando el destino no esta en la red local, lo que obliga al Host a enviar el Datagrama a algún Router intermedio. Es necesario el uso de mascaras de subred para saber si el Host destino de un Datagrama esta o no dentro de la misma red física.

Encaminamiento con Salto al Siguiente.

La forma más común de enrutamiento requiere el uso de una ***Tabla de Enrutamiento IP***, presente tanto en los Host como en los Routers. Estas tablas no pueden tener información sobre cada posible destino, de hecho, esto no es deseable. En ves de ello se aprovecha el esquema de direcionamiento IP para ocultar detalles acerca de los Host individuales, además, las tablas no contienen rutas completas, sino solos la dirección del siguiente paso en esa ruta.

En general una tabla de encaminamiento IP tiene pares (Destino, Router), donde destino es la dirección IP de un destino particular y Router la dirección del siguiente Router en el camino hacia destino. Nótese que Router debe ser accesible directamente desde la maquina actual.

Este tipo de encaminamiento trae varias consecuencias, consecuencia directa de su naturaleza estática:

1. Todo tráfico hacia una red particular toma el mismo camino, desaprovechando caminos alternativos y el tipo de tráfico.
2. Solo el Router con conexión directa al destino sabe si este existe o esta activo.
3. Es necesario que los Routers cooperen para hacer posible la comunicación bidireccional.

Algoritmo de Enrutamiento IP

Ruta Datagrama (Datagrama) {

Extrae de la Cabecera de Datagrama la dirección de destino D;

Extrae de D el prefijo de Red N;

Si N corresponde a cualquier dirección directamente conectada **Entonces**

Envía el Datagrama a D sobre la Red N;

Sino

Si en la tabla hay una ruta especifica para D **Entonces**

Envía Datagrama al salto siguiente especificado;

Sino

Si En la tabla hay una ruta para la red N **Entonces**

Envía Datagrama al salto siguiente especificado;

Sino

Si En la tabla hay una ruta por defecto **Entonces**

Envía el Datagrama a la dirección por defecto;

Sino

Declarar Fallo de Enrutamiento;

Fsi

Fsi

Fsi

Fsi

}

3.4 Subnet Addressing

Para dividir la red en subredes se define una *máscara* en la que están a 1 los bits de la dirección que corresponden a la red-subred, y a cero los que corresponden al host. Se pueden hacer divisiones que no correspondan con bytes enteros, por ejemplo si la máscara fuera 255.255.252.0 se estarían reservando los primeros 6 bits del campo host para la subred y dejando 10 para el host; con lo que podría haber hasta 64 redes con 1024 hosts cada una.

Al crear subredes hay dos direcciones de cada subred que quedan automáticamente reservadas: las que corresponden al campo host todo a 0 y todo a 1; estas se emplean para la designación de la subred y para la dirección broadcast, respectivamente.

Del mismo modo que los valores todo ceros o todo unos del campo host están reservados con un significado especial, el valor todo ceros y todo unos del campo subred también son especiales. El valor

todo ceros se utiliza para representar la subred misma. Por otro lado, el campo subred todo a unos tampoco debería utilizarse porque de lo contrario el significado sería ambiguo, significaría tanto broadcast en la subred como en la red entera.

Mientras que la restricción de no utilizar direcciones todo ceros o todo unos en el campo host se cumple siempre, existen muchas instalaciones que por conveniencia incumplen la restricción equivalente en el campo subred, es decir, consideran válida la subred especificada por el campo subred todo a ceros o todo a unos. Esta práctica se conoce como **subnet-zero** y se adopta para poder aprovechar mejor el espacio de direcciones disponible; con subnet-zero es posible por ejemplo dividir una red clase B o clase C por la mitad en dos subredes mediante la máscara 255.255.128.0 o 255.255.255.128, cosa que no sería posible si no se permitiera esta pequeña 'infracción'.

Para terminar de clarificar el uso de máscaras para crear subredes resumimos en la tabla siguiente las posibles subredes y máscaras que se pueden utilizar con una red clase B:

Bits de subred	Número de subredes	Nº subredes (subred cero)	Bits de host	Número de hosts	Máscara
0	0	0	16	65534	255.255.0.0
1	0	2	15	32766	255.255.128.0
2	2	4	14	16382	255.255.192.0
3	6	8	13	8190	255.255.224.0
4	14	16	12	4094	255.255.240.0
5	30	32	11	2046	255.255.248.0
6	62	64	10	1022	255.255.252.0
7	126	128	9	510	255.255.254.0
8	254	256	8	254	255.255.255.0
9	510	512	7	126	255.255.255.128
10	1022	1024	6	62	255.255.255.192
11	2046	2048	5	30	255.255.255.224
12	4094	4096	4	14	255.255.255.240
13	8190	8192	3	6	255.255.255.248
14	16382	16384	2	2	255.255.255.252
15	32766	32768	1	0	255.255.255.254
16	65532	65536	0	0	255.255.255.255

Tabla 5.8.- Subredes y máscaras que pueden definirse en una red clase B

En el caso de una clase C sería según se recoge en la siguiente tabla:

Bits de subred	Número de subredes	Nº subredes (subred cero)	Bits de host	Número de hosts	Máscara
0	0	0	8	254	255.255.255.0
1	0	2	7	126	255.255.255.128
2	2	4	6	62	255.255.255.192
3	6	8	5	30	255.255.255.224
4	14	16	4	14	255.255.255.240
5	30	32	3	6	255.255.255.248
6	62	64	2	2	255.255.255.252
7	126	128	1	0	255.255.255.254
8	254	256	0	0	255.255.255.255

Tabla 5.9.- Subredes y máscaras que pueden definirse en una red clase C

La funcionalidad de poder dividir una red en subredes de diferentes tamaños se conoce comúnmente como **máscaras de tamaño variable**.

3.5 Subnet Mask

Un host también require saber cuántos bits están siendo usados por la subnet ID y cuántos por el host ID. Esto se obtiene usando la subnet mask que es un valor de 32 bits donde los bits a 1 identifican la red y la subred, y los bits a 0 identifican al host.

Dado la dirección IP y la subnet mask, un host puede saber si un datagrama IP es destinado para (1) un host dentro de su propia subred, (2) un host en otra subred dentro dela misma red, o (3) un host en otra red.

3.8 ifconfig Command

El ifconfig command es normalmente corrido al mismo tiempo para configurar la interface en un host.

A continuación se muestra un ejemplo:

sun % **/usr/etc/ifconfig -a** *SunOS -a option says report on all interfaces*
leO : flags=63<UP, BROADCAST, NOTRAILERS, RUNNING>
inet 140.252.13.33 netmask fffffe0 broadcast 140.252.13.63
slO : flags=105KUP, POINTOPOINT, RUNNING, LINKO>
inet 140.252.1.29 -> 140.252.1.183 netmask ffffff00
loO: flags=49<UP,LOOPBACK,RUNNING>
inet 127.0.0.1 netmask ff000000

La interface lookback es considerada una interface de red.

La bandera LINKO para la interface SLIP es la opción de configuración que habilita la compressed slip (CSLIP).

3.9 netstat Command

El netstat command also prove nformación acerca de las interfaces sobre un sistema. La bandera –I muestra la información de l ainterface, y la bandera -n muestra las direcciones IP en lugar de los nombres de hosts.

sun % **netstat –in**

Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Collis	Queue
leO	1500	140.252.13.32	140.252.13.33	67719	0	92133	0	1	0
slO	552	140.252.1.183	140.252.1.29	48035	0	54963	0	0	0
loO	1536	127.0.0.0	127.0.0.1	15548	0	15548	0	0	0

Este commando muestra la MTU de cada interface, el número de paquetes de entrada, errores de entrada, paquetes de salida, errores de salida, colisiones, y el tamaño concurrente de la pérdida en la salida.

ARP: Address Resolution Protocol

4.1 Introduction

En redes punto a punto el nivel de red (los routers) se ocupa de hacer llegar los datagramas a la red de destino, de acuerdo con rutas perfectamente especificadas en las tablas. Sin embargo dentro de una red broadcast (normalmente una LAN) hace falta un mecanismo que permita descubrir a que dirección MAC corresponde la dirección IP del paquete que se quiere entregar. Esto no puede hacerse mediante una tabla estática, ya que en redes grandes sería muy difícil de mantener. Además, dicha correspondencia

puede cambiar, por ejemplo si a un ordenador se le cambia la tarjeta LAN o si se cambia un servidor a otro ordenador. En estos casos cambia la dirección MAC pero se quiere mantener la dirección IP.

Para permitir la configuración automática de correspondencias dirección MAC-dirección IP se creó el protocolo denominado ARP (Address Resolution Protocol).

4.3 ARP Cache

Las direcciones resueltas vía ARP se mantienen un cierto tiempo en cache; por tanto si Luis termina su sesión vía telnet pero olvidó leer el correo electrónico podrá durante varios minutos establecer otra conexión telnet contra el mismo servidor, sin necesidad de enviar otro paquete broadcast ARP. En realidad cuando se manda una trama ARP *todas* las máquinas de la red, no sólo la destinataria, aprovechan este mensaje para 'fichar' al emisor, anotando en su cache la asociación dirección IP- dirección LAN. De esta forma si mas tarde necesitan contactar con dicha máquina podrán hacerlo directamente, sin necesidad de enviar antes una trama ARP broadcast.
Las entradas ARP cache expiran pasados unos minutos, para permitir que cambie la correspondencia dirección MAC-dirección IP, por ejemplo por avería de una interfaz LAN o cambio de la dirección IP de un host.

Se puede examiner el ARP cache con el commando arp. La opción –a muestra todas las entradas en el cache:

```
bsdi % arp -a
sun
(140.252.13.33)
at
8:0:20:3:f6:42
svr4
(140.252.13.34)
at
0:0:c0:c2:9b:26
```

Las direcciones Ethernet de 48 bits son mostradas como 6 numeros hexadecimales separados por dos puntos.

4.4 ARP Packet Format

Los primeros dos campos en el header Ethernet sonlas direcciones Ethernet origin y destino. La dirección destino especial de Ethernet con todos los bits a 1 significa una dirección boradcast.

Los 2 bytes del *frame type* de Ethernet especifica el tipod e dato que sigue. Para un ARP request o un ARP reply, este campo es 0x0806.

El campo *hard type* especifica el tipo de direcciín de hardware. Este valor es 1 para Ethernet. El tipo *Prot* especifica el tipo de dirección de protocolo que esta siendo mapeada. Este valor es 0x0800 para direcciones IP.

Los próximos dos campos de 1 byte, *hard size* y *prot size,* especifican el tamaño en bytes de las direciones de hardware y de protocolo..

El campo *op* especifica si la operacración es un ARP request (valor a 1), ARP reply (2), RARP request (3), o RARP reply (4). Este campo es requerido desde que el campo *frame type* es el mismo par un ARP request y un ARP reply.

Los 4 campos siguientes son la dirección de hardware del trasmisor (una dirección Ethernet), la dirección del protocolo del trasmisor (una dirección IP),la etiqueta de la dirección de hardware y de protocolo.

4.4 arp Command

El administrador puede especificar la opción –d para eliminar una entrada desde el ARP cache.

Pueden adicionarse entradas con la opción –s. Este require un nombre de host y una dirección Ethernet. La dirección IP corresponde al *hostname,* y la dirección Ethernet son adicionadas al cache. Esta entrada es creada permanentemente bajo la palabra clave temp que aparece en el final de la línea de comando.

RARP: Reverse Address Resolution Protocol

5.1 Introduction

A veces se plantea el problema inverso a ARP, es decir hallar la dirección IP de una determinada dirección LAN. Para esto se inventó RARP (Reverse Address Resolution Protocol), que consiste en que la estación emita una trama broadcast indicando su dirección LAN y solicitando alguien le informe de cual es la dirección IP que le corresponde. En este caso una máquina en la red local (el servidor RARP) atenderá la petición, consultará en sus tablas, y devolverá la dirección IP correspondiente.

5.2 RARP Packet Format

El formato de un paquete RARP es casi identico a un paquete ARP. Las únicas diferencias son que el *frame type* es 0x8035 para un RARP request o reply, y el campo *op* tiene un valor de 3 para un RARP request y 4 para un RARP reply.

Como en ARP, el RARP request es broadcast y el RARP reply es normalmente unicast.

Ping Program

6.1 Introduction

El nombre "ping" es tomado de la operación sonar para localizar objetos. El programa envia un mensaje ICMP echo request hacia un host, esperando que un ICMP echo reply sea retornado.

Cuando no se puede hacer Ping hacia un host, no debe estar habilitado el TELNET o el FTP hacia ese host, y se puede determinar que existe un problema. Ping también hace un round-trip time hacia el host, dandonos una indicación de cuan lejos esta ese host.

6.2 Ping Program

El número de secuencia comienza en 0 y es incrementado cada vez que un nuevo echo request es enviado. Ping muestra el número secuencial de cada paquete retornado, permitiendo saber si los paquetes estan perdidos, reordenados o duplicados.

6.3 IP Record Route Option

El programa ping da la oportunidad de observar la opción record route (RR) de IP. Muchas versiones de ping nos permiten manejar la opción -R para ver las características del record route. Cuando el datagrama llega al destino final, la lista de direcciones IP deberán ser copiadas dentro de la salida ICMP echo reply, y todos los routers sobre la ruta de regreso también adicionarán sus direcciones IP a la lista. Cuando ping recibe el echo reply mostrará la lista de direcciones IP por las que pasó.

Traceroute Program

7.1 Introduction

El programa Taceroute nos muestra la ruta que los datagramas IP siguen desde un host hacsta otro, ya que los datagramas IP no siguen una misma ruta para llegar al mismo destino.

7.2 Traceroute Program Operation

Traceroute usa ICMP y el campo TTL en el header IP header. El campo TTL d (time-to-live) es un campo de 8 bits que el trasmisor inicializa con algún valor.

Este campo TTL se va decrementando conforme vaya dando saltos para llegar a su destino, el objeto de esto es prevenir que un datagrama este en un loop infinito.

Traceroute envia datagramas UDP al host destino, pero este escoge el número de Puerto UDP del destino para ser valor único y haciendo imposible que una aplpicación en el host destino este usando ese puerto.

7.3 LAN Output

A continuación mostramos un ejemplo de lo que ocurre al utilizar traceroute para ir a un host slip

svr4 % **traceroute slip**
traceroute to slip (140.252.13.65), 30 hops max. 40 byte packets
1 bsdi (140.252.13.35) 20 ms 10 ms 10 ms
2 slip (140.252.13.65) 120 ms 120 ms 120 ms

La primera linea de salida sin numeración da el nombre y la dirección IP del destino e indica que traceroute incremento el TTL hasta 30. El tamaño del datagrama de 40 bytes permite ver que 20 bytes son del header IP, 8 bytes del header UDP y 12 bytes usados para datos.

Las próximas dos lineas comienzan con el TTL, seguidas por el nombre del host o router,y la dirección IP. Por cada valor de TTL son enviados tres datagramas. En esta salida los primeros tres datagramas tienen un TTL de 1 y los mensajes ICMP fueron retornados en 20, 10, y 10 ms. Desde que el TTL 2 rechazo el destino final el programa se detuvo.

7.4 WAN Output

A continuación mostraremos un ejemplo acerca de una salida WAN utilizando traceroute

sun % **traceroute nic.ddn.mil**
traceroute to nic.ddn.mil (192.112.36.5), 30 hops max, 40 byte packets

1 netb.tuc.noao.edu (140.252.1.183) 218 ms 227 ms 233 ms
2 gateway.tuc.noao.edu (140.252.1.4) 233 ms 229 ms 204 ms

3 butch.telcom.arizona.edu (140.252.104.2) 204 ms 228 ms 234 ms
4 Gabby.Telcom.Arizona.EDU (128.196.128.1) 234 ms 228 ms 204 ms
5 NSIgate.Telcom.Arizona.EDU (192.80.43.3) 233 ms 228 ms 234 ms

6 JPLI.NSN.NASA.GOV (128.161.88.2) 234 ms 590 ms 262 ms
7 JPL3.NSN.NASA.GOV (192.100.15.3) 238 ms 223 ms 234 ms
8 GSFC3.NSN.NASA.GOV (128.161.3.33) 293 ms 318 ms 324 ms
9 GSFC8.NSN.NASA.GOV (192.100.13.8) 294 ms 318 ms 294 ms
10 SURA2.NSN.NASA.GOV (128.161.166.2) 323 ms 319 ms 294 ms
11 nsn-FIX-pe.sura.net (192.80.214.253) 294 ms 318 ms 294 ms
12 GSI.NSN.NASA.GOV (128.161.252.2) 293 ms 318 ms 324 ms

13 NIC.DDN.MIL (192.112.36.5) 324 ms 321 ms 324 ms

7.5 IP Source Routing Option

Generalmete IP routing es dinámico tomando la desición del próximo salto por el cual el router enviará el datagrama. Las aplicaciones no tiene control sobre esto, por lo que se utilize Traceroute para mostrar lo que el router esta haciendo.

La idea detrás del source routing es que el transmisor especifique la ruta. Existen dos formas:

- *Strict* source routing. El transmisor especificará la ruta exacta por la que el datagrama IP viajrá. Si el router encuentra un salto que no esta en la fuente enviará un ICMP error de regreso.
- *Loose* source routing. El transmisor especificará una lista de direcciones IP por las cuales podrá ir el datagrama pudiendo este escoger el próximo salto dentro de la lista.

IP Routing

8.1 Introducción

El routeo es una de las principales funciones de IP. Los datagramas a ser ruteados pueden ser generados en el localhost o en otro host. En el ultimo caso este host dbe ser configurado para actuar como router, o los datagramas recibidos a través de la interface de red no se han caido.

Los demonios más comunes utilizados sobre los sistemas UNIX son los programas routed y gated. (El término demonio significa que el proceso está corriendo “en background". Los demonios empiezan cuando el sistema arranca y siguen hasta que el sistema se detiene.)

8.2 Routing Principles

Lo principal en el IP routing es comprender que es mantenido en el kernel por las tablas de ruteo. Aqui se encuentran todas las decisions de ruteo tomadas por IP.

1. Buscar la direccion del host.
2. Buscar la dirección de red.
3. Buscar una entrada por default.

El ruteo hecho por IP al buscar en la tabla de rutoe y decide por que interface será enviado el paquete se conoce como meanismo de ruteo.

Simple Routing Table

Observamos una tabla de rutoe utilizando el commando netstat con ls opciones -rn

svr4 % **netstat -rn**
Routing tables

Destination	Gateway	Flags	Refcnt	Use	Interface
140.252.13.65	140.252.13.35	UGH	0	0	emd0
127.0.0.1	127.0.0.1	UH	1	0	lo0
default	140.252.13.33	UG	0	0	emd0
140.252.13.32	140.252.13.34	U	4	25043	emd0

Hay 5 diferentes banderas que s muestran para una ruta dada.

U La ruta esta levantado.

G La ruta es para una gateway (router).

H La ruta es para un host, that is, the destination is a complete host address.

D La ruta fue creada por una redirección

M La ruta fue modificada por una redirección.

La bandera G es imprtante porque diferencia entre na ruta directa y una indirecta. La bandera H especifica que la dirección de host esta completa.

La complejidad de las tables de ruteo del host dependen de la topología de red a la cual el host tiene acceso.

1. l Cuando el host no está conectado a ninguna red. Los protocolos TCP/IP podrán ser usados pero solo para comunicarse con el mismo.
2. Otro caso es un host conectado a una LAN simple, solo puede acceder a host sobre la misma LAN.
3. Otro caso es cuando otras redes como internet estan conectadas a través de un simple router.
4. El ultimo caso es cuando un host o red específica son adidionados(asignación estática).

8.3 ICMP Host and Network Unreachable Errors

El ICMP "host unreachable" error message es enviado por un router cuando este recibe un datagrama IP que no puedi descifrar o reenviar.

8.4 ICMP Redirect Errors

El ICMP redirect error es enviado por el router al transmisor del datagrama IP cuando este debería ser enviado a un router diferente.

Se asume que el host envia el datagrama IP al RI. Esta decision de ruteo es generalmente tomada porque el RI es el router por default del host.

1. RI recibe el datagrama y ejecuta un lookup en su tabla de ruteo y determina que R2 es el correcto próximo salto para ser enviado el datagrama.
2. RI envia un ICMP redirect al host, diciendole que el datagrama será enviado hacia R2.

8.5 ICMP Router Discovery Messages

Router Operation

Cuando un router esta levantado trasmite advertencies periodicas sobre las interfaces capaces de broadcasting o multicasting. Estas advertencias no son exactamente periodicas, pero son randómicas, para reducer la posibilidad de sincronización con otro router sobre la misma subnet. El intervalo de tiempo que se toma entre cada advertencia esta entre 400 a 600 segundos.

Host Operation

Un host envia tres solicitudes al router, cada 3 segundos y cuando una advertencia válida es recibida las solicitudes se detienen. Un host también escucha advertencias de los routers adyacentes. Mientras el router por default este levantado, ese router enviará advertencias cada 10 minutos, con un tiempo de vida de 30 minutos.

Implementación

Los router discovery messages son generados y procesados por un proceso usuario (un demonio). El demonio deberá estar configurado para actuar como router o host.

Dynamic Routing Protocols

9.1 Introducción

Las entradas de la tabla de rutoe son creadas por default cuando se configura una interface, adicionadas por el commando route, o creadas por un ICMP redirect. Esto esta bien si la red es pequeña, caso contrario se utiliza el ruteo dinámico.

9.2 Dynamic Routing

Dynamic routing ocurre con los routers se comuncan con routers adyacentes, informando a los demas a que redes estan conectados Los routers deben comunicarse utilizando un *routing protocol*. El proceso sobre el que el router esta corriendo el routing protocol, comunicandose con los routers cercanos, es llamado un *routing daemon.*

El uso de ruteo dinámico no cambia la amnera en que el kernel realiza el ruteo de la capa IP. Si el demonio encuentra varios routers para un destino entonces escoge el major y lo adiciona a la tabla de rutoe del kernel. Cuando el demono detecta que un router se ha caido inmediatamente busca una vía alternativa para solucionar el problema y elimina el router caido de la tabla del kernel

9.3 Routing Daemons

La siguiente tabla compara diversos routing protocols soportados para el ruteo y dos diferentes versiones de gated.

Daemon	Interior Gateway Protocol			Exterior Gateway Protocol	
	HELLO	RIP	OSPF	EGP	BGP
Routed		V1			
gated, Version 2	*	V1		*	V1
gated, Version 3	*	V1, V2	V2	*	V2, V3

Routing protocols soportados para ruteo y gated.

9.4 RIP: Routing Information Protocol

RIP es uno de los protocolos de routing más antiguos, derivado del protocolo de routing de XNS (Xerox Network Systems); RIP sufre los problemas típicos de los algoritmos basados en el vector distancia, tales como la cuenta a infinito, envío de excesiva información de routing, etc. Estos problemas aumentan a medida que aumenta el tamaño de los sistemas autónomos. A pesar de esto es aún muy utilizado en la Internet. Existen dos versiones de RIP: la versión 1, que se definió en el RFC 1058 y se publicó en 1983 (aunque se empezó a utilizar mucho antes) se ha declarado histórica, es decir su uso está desaconsejado. En vista de la popularidad de RIP y de los muchos problemas que presentaba en 1993 se publicó RIP versión 2 intentando resolver algunos de ellos (RFC 1388).

Opración normal

A continuación estan los pasos que sigue una operación normal de rutoe usando RIP. El Puerto utilizado por RIP es UDP 520.

- Initialization.
- Request received.
- Response received.

- Regular routing updates.
- Triggered updates.

9.5 CIDR: Classless Interdomain Routing

El rápido crecimiento de la Internet está creando varios problemas, el más importante de los cuales es el rápido agotamiento de las direcciones. Las redes clase A ya prácticamente no se asignan, y resulta muy difícil obtener una clase B. Muchas empresas no tienen bastante con una clase C pero una B les resulta excesiva. Entre las medidas que se están adoptando para paliar el problema de escasez de direcciones está la de asignar un conjunto de redes clase C donde antes se asignaba una clase B. De esta forma se puede ajustar mejor el rango de direcciones asignado a las necesidades reales previstas de cada organización

Esto ha resuelto un problema pero ha creado otro: el crecimiento desmedido de las tablas en los routers. Muchos routers de la Internet funcionan con unas tablas de encaminamiento que solo recogen una pequeña parte de todas las redes existentes, y disponen de una ruta por defecto por la cual se sale hacia el resto de la Internet. Sin embargo, a medida que nos aproximamos al centro o 'backbone' de la Internet las tablas tienen que ser necesariamente mas exhaustivas; en las tablas de los denominados 'core routers' se mantienen entradas para la mayoría de las redes existentes en la Internet.

Para solucionar este problema se adoptó en 1993 un sistema denominado CIDR (Classless InterDomain Routing) descrito en el RFC 1519. Se trata de dos medidas complementarias:

La primera consiste en establecer una jerarquía en la asignación de direcciones. Antes de CIDR la asignación de números de red se hacía por orden puramente cronológico, independientemente de la ubicación geográfica, lo cual equivalía en la práctica a una asignación aleatoria del número de red. Con CIDR se han asignado rangos por continentes:

194.0.0.0 a 195.255.0.0 para Europa
198.0.0.0 a 199.255.0.0 para Norteamérica
200.0.0.0 a 201.255.0.0 para Centro y Sudamérica
202.0.0.0 a 203.255.0.0 para Asia y la zona del Pacífico

A su vez dentro de cada uno de estos rangos se ha dado una parte a cada país, y dentro de éste un rango a cada proveedor de servicios Internet. Con esta distribución regional de los números y los cambios pertinentes en el software las entradas en las tablas de routing pueden agruparse, con lo que las tablas se simplifican.

La segunda medida adoptada por CIDR es en realidad es un caso particular de la anterior. Consiste en dar a cada organización (bien directamente o a través de su proveedor correspondiente) un conjunto de redes clase C ajustado a lo que son sus necesidades previstas, dándole siempre un rango contiguo y un número de redes que sea potencia entera de 2 (es decir 1, 2, 4, 8 redes, etc.) elegidas de modo que tengan una máscara común en la parte de red.

Obsérvese que CIDR es en realidad el mismo mecanismo que las subredes, pero aplicado en sentido inverso. Las subredes permiten dividir una red, ampliando la parte red a costa de la parte host de la dirección. El CIDR funde diferentes redes en una, reduciendo la parte red y ampliando la parte host. Por este motivo el CIDR también se conoce como *supernet addressing*.

Cuando se utiliza CIDR los hosts finales que se encuentran en distintas redes del mismo tipo no pueden hablar directamente entre ellos, han de hacerlo a través de un router, a menos que soporten CIDR que no es lo normal en hosts.

UDP: User Datagram Protocol

10.1 Introducción

El protocolo UDP de Internet es un servicio no orientado a conexión. Entre las aplicaciones que utilizan UDP se encuentran TFTP (Trivial File Transfer Protocol), DNS (Domain Name Server), SNMP (Simple Network Management Protocol), NTP (Network Time Protocol) y NFS (Network File System), etc.

Las TPDUs intercambiadas por UDP se denominan *mensajes* o *datagramas UDP*, lo cual da una idea de la naturaleza del protocolo. Debemos decir que el valor 17 en el campo protocolo del datagrama IP identifica un mensaje UDP.

Una característica importante de UDP es que puede ser utilizado por aplicaciones que necesitan soporte de tráfico multicast o broadcast. Con TCP esto no es posible debido a la naturaleza punto a punto, orientada a conexión del protocolo.

10.2 UDP Header

El ancho del campo UDP es el ancho del UDP header y UDP data en bytes. El valor mnimo en este campo es 8 bytes. (se puede enviar un datagrama UDP con 0 bytes) Este ancho es redundante. El ancho total del datagrama UDP es el minimo que acepta el hedear de IP.

10.3 UDP Checksum

El uso de este campo en UDP es opcional; cuando se envía información en tiempo real (audio o vídeo digitalizado) su uso puede omitirse. Para el cálculo se aplica el mismo algoritmo que en TCP (suma complemento a 1 de todo el mensaje dividido en campos de 16 bits, y complemento a 1 del resultado). En el cálculo se utiliza todo el mensaje, incluida la cabecera y se antepone una pseudocabecera igual a la utilizada en TCP (con la dirección IP de origen, de destino y la longitud del mensaje) de forma que se verifica que sean correctos los datos y la información de control (puertos, direcciones IP, etc.) incluidos los datos fundamentales de la cabecera IP. Si la verificación del checksum en el receptor da error el mensaje es simplemente descartado sin notificarlo al nivel de aplicación ni al emisor.
Datos contiene los datos a transmitir. Un mensaje UDP ha de estar contenido necesariamente en un datagrama IP, lo cual fija la longitud máxima de este campo.

TCP: Transmission Control Protocol

11.1 Introducción

El intercambio de segmentos en TCP se desarrolla de acuerdo con un protocolo de ventana deslizante, con un número de secuencia de 32 bits. El número de secuencia cuenta bytes transmitidos, por lo que la secuencia se reinicia cada 4 GB transmitidos (equivalentes a 4,3 minutos en una línea ATM de 155 Mbps, suponiendo que toda la capacidad útil de la línea se utilizara para una esa conexión TCP).
TCP utiliza ACK piggybacked, por lo que en el campo de cabecera de cada segmento hay previstos dos campos de 32 bits, uno para el número de secuencia y otro para el número de ACK. El campo número de secuencia indica el número del primer byte transmitido dentro de ese segmento. El campo ACK indica el número del primer byte que se espera recibir en el siguiente segmento (o sea, el siguiente byte que se espera recibir, siguiendo el estilo del campo 'next' en HDLC). En la práctica el ACK piggybacked no se aprovecha casi nunca ya que la mayoría de las aplicaciones están diseñadas de forma que la información se envía de forma asimétrica; normalmente es necesario que el TCP receptor envíe periódicamente segmentos vacíos con el único fin de informar al emisor que los datos han sido recibidos.
El mecanismo normal de funcionamiento de TCP es retroceso n, aunque también puede utilizarse repetición selectiva si las dos entidades participantes lo soportan.

11.2 TCP Services

TCP provee confiabilidad debido a lo siguiente:

- Cuando un dato es fragmentado TCP considera el mayor pedazo para ser enviado. Este es totalmente diferente de UDP, la unidad de información pasada de TCP a IP es llamada segmento.
- Cuando TCP envia un dato este contiene un timer, esperando del destinatario un ACK. Si no es retransmitido un ACK en un lapso determinado este datos es retrasmitido.
- Cuando TCP recibe datos del otro lado de la conexión, este envia un ACK. Este ACK normalmente demora una fracción de Segundo en ser enviado.
- TCP mantiene un checksum sobre su cabecera y sus datos.
- Como los datoagramas de TCP son transmitidos mediante IP y este los envie en cualquier orden, existe en el destino una aplicación TCP que los reordena para procesarlos.
- Como los datagramas IP pueden esatr duplicados, TCP en el destino se encarga de descartarlos.
- TCP también prove control de flujo.

11.3 TCP Header

La cabecera de un segmento TCP tiene la siguiente estructura:

Campo	**Longitud (bits)**
Puerto origen	16
Puerto destino	16
Número de secuencia	32
Número de ACK	32
Longitud de cabecera TCP	4
Reservado	6
URG (Urgent)	1
ACK (Acknowledgement)	1
PSH (Push)	1
RST (Reset)	1
SYN (Synchronize)	1
FIN (Finish)	1
Tamaño de ventana	16
Checksum	16
Puntero de datos urgentes	16
Opciones	0, 32, 64, ...
Datos	0-524120 (65515 bytes)

Puerto origen y *puerto destino* identifican los puertos que se van a utilizar en cada host para comunicar con las aplicaciones que intercambian datos.

Número de secuencia indica el número de secuencia que corresponde en la conexión al primer byte que se envía en el campo datos.

Número de ACK indica el número de secuencia del próximo segmento que se espera recibir del otro lado.

Longitud de cabecera TCP especifica la longitud en palabras de 32 bits, excluido el campo datos (el campo opciones hace que dicha longitud pueda variar).

A continuación hay 6 bits no utilizados, seguidos por seis indicadores de un bit cada uno:

- URG (urgent): sirve para indicar que el segmento contiene datos urgentes; en ese caso el campo puntero de datos urgentes contiene la dirección donde terminan éstos.

- ACK (acknowledgement): indica que en este segmento el campo *Número de ACK* contiene el número del próximo byte que se espera recibir. En la práctica el bit ACK esta a 1 siempre, excepto en el primer segmento enviado por el host que inicia la conexión.

- PSH (push): indica que el segmento contiene datos PUSHed, es decir, que deben ser enviados rápidamente a la aplicación correspondiente, sin esperar a acumular varios segmentos.

- RST (reset): se usa para indicar que se ha detectado un error de cualquier tipo; por ejemplo una

- SYN (synchronize): está puesto sólo en el primer mensaje enviado por cada lado en el inicio de la conexión.

- FIN (finish): indica que no se tienen más datos que enviar y que se quiere cerrar la conexión. Para que una conexión se cierre de manera normal cada host ha de enviar un segmento con el bit FIN puesto.

Tamaño de ventana indica la cantidad de bytes que se está dispuesto a aceptar del otro lado en cada momento. Se supone que se garantiza una cantidad suficiente de espacio en buffers. Mediante este parámetro el receptor establece un control de flujo sobre el caudal de datos que puede enviar el emisor.

Checksum sirve para detectar errores en el segmento recibido; estos podrían ser debidos a errores de transmisión no detectados, a fallos en los equipos (por ejemplo en los routers) o a problemas en el software (por ejemplo reensamblado incorrecto de fragmentos).

Puntero de datos urgentes indica el final de éstos, ya que el segmento podría contener datos no urgentes. TCP no marca el principio de los datos urgentes, es responsabilidad de la aplicación averiguarlo.

El campo opciones habilita un mecanismo por el cual es posible incluir extensiones al protocolo. Entre las más interesantes se encuentran las siguientes:

- Tamaño máximo de segmento

- Uso de repetición selectiva (en vez de retroceso n)

- Uso de NAK (acuse de recibo negativo en caso de no recepción de un segmento)

- Uso de ventana extendida (mayor de 64 KBytes)

11.6 Timeout of Connection Establishment

Existen instancia severas cuando la conexión no puede ser establecida. En la siguiente tabla se muetra el tcpdump output cuando no se puede establecer una conexión.

1	0.0	bsdi-1024 > svr4.discard: win 4096 <mss 1024> [tos 0x10]	S	291008001:291008001(0)
2	5.814797 (5.8148)	bsdi-1024 > svr4.discard: win 4096 <mss 1024> [tos 0x10]	S	291008001:291008001(0)
3	29.815436 (24.0006)	Bsdi.l024 > svr4.discard: win 4096 <mss 1024> [tos 0x10]	S	291008001:291008001(0)

tcpdump output para establecer una conexión con time out.

El punto interesante en esta salida es cuan frecuentemente el TCP del cliente envia un SYN para tratar de establecer la conexión. El segundo segmento es enviado 5.8 segundos después del primero,y el tercero es enviado 24 segundos después del segundo.

11.7 Maximum Segment Size

El mámimo tamaño de segmento (MSS) es el largo "chunk" del dato que TCP enviará hacia el otro punto. Cuando una conexión es etablecida, cada destino puede saber el tamaño del MSS que esta recibiendo. El datagrama IP resultante tiene un tamaño de 40 bytes: 20 bytes para el header de TCP y 20 bytes para el headre IP.

11.8 Round-Trip Time Measurement

Para estimar un valor razonable del timer de retransmisión TCP mide lo que tardan en llegar los ACK de los segmentos enviados; se supone que estos tiempos son una buena estimación del tiempo de ida y vuelta o RTT (Round Trip Time) de los segmentos, en base al cual ha de calcularse el valor del timer de retransmisión. El valor medio del RTT (que denominaremos MRTT) se estima mediante la fórmula iterativa:

$$MRTT = \alpha\ MRTT_{viejo} + (1-\alpha)\ RTT$$

donde RTT es el tiempo de ida y vuelta medido para el último ACK recibido. El parámetro α permite ajustar el peso o la importancia que se quiere dar al último valor frente a los anteriores; con un α pequeño se consigue que los valores anteriores tengan poca relevancia, adaptándose así a situaciones cambiantes con rapidez. Con α grande se reacciona más lentamente; de esta forma puede ajustarse el grado de 'inercia' de MRTT. En TCP α vale normalmente 7/8

Obtener una buena estimación de MRTT resuelve sólo la primera parte del problema; sabemos que los valores de RTT se distribuirán alrededor del valor medio, pero cual es el valor adecuado del timer de retransmisión, o sea, cual es el valor a partir del cual podemos considerar que el ACK no llegará?. Las primeras implementaciones de TCP utilizaban 2*MRTT como valor del timer, pero eso tenía el inconveniente de que en unos casos los valores fluctuaban mucho y el valor 2*MRTT resultaba demasiado bajo y se producían excesivas retransmisiones innecesarias; en otros casos la dispersión era pequeña y 2*MRTT daba un valor excesivo ya que con un valor menor se podía considerar perdido un segmento en la mayoría de los casos. Si queremos una estimación más aproximada del timeout necesitamos saber, además del valor medio, el grado de dispersión o anchura de la campana, es decir la desviación estándar. Para estimar esta magnitud de manera sencilla se utiliza la siguiente fórmula:

$$D = \beta\ D_{viejo} + (1-\beta)\ |MRTT_{viejo} - RTT|$$

Donde β suele valer 3/4. Una vez obtenidos MRTT y D podemos calcular un valor de timeout mas adecuado para cada situación. Generalmente se utiliza la fórmula:

$$\text{Timeout de retransmisión} = MRTT + 4 * D$$

SNMP: Simple Network Management Protocol

12.1 Introducción

La administración de red de TCP/IP consiste de estaciones de administración de redes (managers) comunicándose con los elementos de red. Los elementos de red pueden ser cualesquiera que corran bajo el conjunto de protocolos TCP/IP: hosts, routers, X terminals, etc.El software en el elemento de red que corre el software adminitrador es llamado el agente.

"La comunicación puede ser en dos vias: el administrador pregunta al agente por un valor específico o el agente le dice al administrador sobre algo importante que haya ocurrido. El administrador de red TCP/IP consiste de tres piezas:

1. Un *Management Information Base* (MIB)
2. Un set de estructuras comunes y un esquema de identificación para referirse a las variables en el MIB. Esto se llama *Structure of Management Information* (SMI).
3. El protocolo entre el administrador y el elemento, llamado *Simple Network Management Protocol* (SNMP). Este detalla el formato de los paquetes intercambiados.

12.2 Protocol

SNMP define solo 5 tipos de mensajes que son intercambiados entre el administrador y el agente.

1. Traer el valor de una o más variables: el get-request operator.

2. Traer la próxima variable después de una o varias variables especificadas: el get-next-request operator.
3. Colocar el valor de una o más variables: el set-request operator.
4. Retornar el valor de una o más variabless: el get-response operator. Este es un mensaje retornado por enl agente al adminstrador en respuesta al get-request, get-next-request, y set-request operators.
5. Notificar al administrador cuando algo sucede en el agente: el trap operator.

Los primeros tres mensajes son enviados desde el administrador al agente, y los dos últimos desde el agente al administrador.

Desde que 4 de los 5 mensajes SNMP son simples protocolos request-reply SNMP usa UDP. El administrador envia estos 3 requests por el puerto UDP 161. El agente envía traps por el puerto UDP 162.

A continuación mostramos los tipos de mensajes:

PDU type	Name
0	get-request
1	get-next-request
2	set-request
3	get-response
4	trap

Tipos de PDU para mensajes SNMP.

12.3 Structure of Management Information

SNMP usa pocos tipos de datos diferentes. A continuación mostramos estos tipos de datos:.

- INTEGER. Algunas variables son declaradas como enteros sin restricciones (ejm. la MTU de una interface), algunas definidas tomando valores específicos (ejm. La bandera IP forwarding esta en 1 si el forwarding es habilitado o 2 si es deshabilitado), y otras definidas con un valor máximo y un mínimo (ejm., UDP y TCP port numbers).
- OCTET STRING. UNa cadena de 0 o más bytes de 8 bits.
- DisplayString. Una cadena de 0 o mas bytes de 8 bits, pero cada byte debe ser un caracter del NVT ASCII set.
- OBJECT IDENTIFIER
- NULL. Indica que la variable correspondiente no tiene valor.
- lpAddress. UNa cadena de octetos de ancho 4, con 1 byte por cada byte de la dirección IP.
- PhysAddress. Una cadena de octetos especificando la dirección física
- Counter. Un entero no negative que se va incrementando desde 0 a 2^{32}-1 (4,294,967,295), y vuelve a 0.
- Gauge. Un entero no negative entre 0 y 2^{32} -1, cuyo valor puede incrementar o decrementar
- TimeTicks.
- SEQUENCE. Es similar a la estructura en el lenguaje de programación C.
- SEQUENCE OF. Esta es la definición de un vector, cuyos elementos tienen los mismos tipos de datos.

12.4 Object Identifiers

Un *object identifier* es un tipo de dato specificado por un bojeto llamado autoritativo. Por "autoritativo" entendemos que no son asignados randomicamente, pero son colocados por alguna organización que tiene la responsabilidad sobre estos identificadores de grupo.

Es una secuencia de enteros separados por puntos decimal. Estos enteros atraviezan una triple estructura, similar al DNS en el sistema de archivos de UNix. Hay una ruta sin nombre al comienzo de los tres donde los object identifiers comienzan.

Cada nodo también tiene un nombre textual. El nombre correspondiente al object identifier 1.3.6.1.2.1 es iso.org.dod.internet.mgmt.mib. Estos nombres son para facilitar su comprensión.

12.5 Introducción al Management Information Base

El *Management Information Base,* o MIB, es la base de datos de la información mantenida por el agente al cual el administrador puede preguntar o disponer. El MIB esta dividido dentro de grupos, interfaces.

En el grupo UDP este es un grupo simle con unas pocas variables y una sola tabla.

Existen 4 variables simples y y una tabla conteniendo dos variables simples. A continuación describimos las 4 variables simples.

Name	Datatype	R/W	Description
udpInDatagrams	Counter		Number of UDP datagrams delivered to user processes.
udpNoPorts	Counter		Number of received UDP datagrams for which no application process was at the destination port.
udpInErrors	Counter		Number of undeliverable UDP datagrams for reasons other than no application at destination port (e.g., UDP checksum error).
udpOutDatagrams	Counter		Number of UDP datagrams sent.

Simple variables in udp group.

UDP listener table, index = < *udpLocalAddress* >.< *udpLocalPort* >			
Name	Datatype	R/W	Description
udpLocalAddress	lpAddress		Local IP address for this listener. 0.0.0.0 indicates the listener is willing to receive datagrams on any interface.
udpLocalPort	[0..65535]		Local port number for this listener.

Variables in udpTable.

Telnet and Rlogin: Remote Login

13.1 Introducción

Remote login es una de las más populares aplicaciones de internet. Podemos acceder remotamente a un terminal en otro lugar.

Provve dos aplicaciones populares:.

1. Telnet es una aplicación estandard que trabaja entre hosts que utilizan diferentes sistemas operativos.
2. Rlogin fue desarrollado para trabajar bajo sistemas UNIX pero ha sido modificado para trabajar bajo diferentes sistemas operativos

13.2 Telnet Protocol

Telnet fue diseñado para trabajar entre varios hosts y varios terminales. El NVT es un dispositivo imaginario que se encuentra en ambos extremos de la conexión, el cliente y el servidor.

NVT ASCII

El término *NVT ASCII* se refiere a la variante de 7 bits de US de los caracteres ASCII que pueden ser usados atraves de los protocolos de internet. Los 7 bits son enviados con un 8 bit, con el bit seteado a 0.

Telnet Commands

Telnet usa una única banda en ambas direcciones. El byte 0xff (255 decimal) es llamado IAC, por "interpret as command." El próximo byte es el byte de comando. Para enviar un byte de datos de 255, son enviados dos bytes consecutivos del mismo tamaño. A continuación se muestra la lista de todos los comandos de Telnet.

Name	Code (decimal)	Description
EOF	236	end-of-file
SUSP	237	suspend current process (job control)
ABORT	238	abort process
EOR	239	end of record
SE	240	suboption end
NOP	241	no operation
DM	242	data mark
BRK	243	Break
IP	244	interrupt process
AO	245	abort output
AYT	246	are you there?
EC	247	escape character
EL	248	erase line
GA	249	go ahead
SB	250	suboption begin
WILL	251	option negotiation
WONT	252	option negotiation
IX)	253	option negotiation
DONT	254	option negotiation
IAC	255	data byte 255

Comandos de Telnet, cuando era precedido por IAC (255).

Routers

14.1 Definición

Los Routers son aquellos que conectan dos o más redes, sirviendo de enlace entre ellas, trabajan en la capa de Internet, encargándose de encaminar o enrutar paquetes de datos entre máquinas de redes diferentes.

Para poder funcionar de esta forma deben pertenecer a cada una de las redes que conectan, como si fueran un host más de las mismas. De esta forma, un router que conecte dos redes debe tener una tarjeta de red diferente para cada una de las redes y, consecuentemente, dos direcciones MAC diferentes. También debe tener asignada una dirección IP en cada una de las dos redes, ya que si no sería imposible la comunicación con las máquinas de las mismas.

El esquema de dos redes conectadas por un router podría ser el representado en la siguiente imagen:

Ahora, cuando un host envía una petición ARP para averiguar la dirección MAC correspondiente a una IP dada y no es respondido por ningún equipo de su red, envía los paquetes correspondientes a un router que tiene configurado para este tipo de envíos, denominado gateway por defecto.

Una vez que el router recibe los paquetes de datos utiliza un parámetro especial, denominado máscara de red, que sumado lógicamente a la dirección IP destino le da la red a la que pertenece el host buscado. Pasa entonces los paquetes a la red a la que pertenece C, haciendo una nueva petición de broadcast preguntando la MAC de C. Este le responde, y entonces el router le envía los paquetes directamente. Si C desea responder a A, el proceso se invierte.

Este proceso es necesario realizarlo sólo una vez, ya que en esta tanto los host A y C como el router anotan las parejas de direcciones MAC-IP en unas tablas especiales, denominadas tablas de enrutamiento, que usarán en envíos de datos posteriores para enrutar los paquetes directamente.

Resumiendo, los routers son los principales responsables de la correcta comunicación entre máquinas de diferentes redes, encargándose en este proceso de enrutar correctamente los paquetes de datos.

Los routers son dispositivos de red que raramente se encuentran aislados entre sí. Al contrario, suelen estar interconectados, formando una especie de "telaraña" que hace posible el tráfico de datos entre redes separadas físicamente.

Tomando como ejemplo la Red de redes, Internet, cuando un ordenador envía una serie de paquetes de datos a otro situado en otra ciudad o país, estos son encaminados de router a router a lo largo del camino entre ambas máquinas. Cada paso de un paquete de un router a otro se denomina "salto", y el principal objetivo de todos y cada uno de los routers que intervienen en la transferencia del paquete es que éste llegue a su destino en el menor número posible de saltos, por la mejor ruta posible.

Para poder realizar esta tarea, los routers se comunican constantemente entre sí, informándose de las rutas bloqueadas, de las máquinas intermedias que se encuentran caídas o saturadas de tráfico, aprendiendo con ello cuál es el router idóneo para enviarle los paquetes recibidos.

Si consideramos ahora el caso de un router segmentando una red local (LAN), aunque ahora no debe enviar los paquetes a otro router, sí que tiene que saber por qué puerto debe enviar los datos para que lleguen a la máquina local destino.

Esta habilidad de "saber" a dónde tienen que enviar los paquetes de datos que reciben la consiguen almacenando en su interior una tabla especial, conocida como tabla de ruteo, en la que van anotando las direcciones IP de las máquinas que se comunican con él y el puerto por el que está accesible esa máquina.

Así, cuando a un router llega un paquete, mira en su tabla de ruteo. Si está en ella referenciada la dirección IP de la máquina destino, también lo estará el puerto por el que ésta es accesible, con lo que envía por él el paquete. En caso de no estar la IP en la tabla, manda una petición de respuesta por todos los puertos, preguntando en cuál de ellos se encuentra la máquina destino, y una vez obtenido el puerto

de acceso, ingresa la nueva pareja IP/PUERTO en su tabla de ruteo, con lo que los próximos paquetes para esa máquina los enviará directamente.

Para evitar mantener en su tabla direcciones IP que hayan quedado obsoletas, cada cierto tiempo borra aquellas que no tienen actividad y las que, tras enviarles paquetes, no han respondido. Esto lo consiguen manteniendo "conversaciones" entre ellos, en unos lenguajes especiales denominados "protocolos de enrutamiento".

Destino	Métrica	Interface
210.100.12.0	0	eth0
210.100.12.5	0	eth0
192.80.26.13	8	s0
135.0.25.124	5	s1

Tabla de enrutamiento simple

14.2 Componentes Básicos de un Router

Básicamente, podemos considerar un router como un ordenador especial que funciona solo en las tres primeras capas de la arquitectura TCP/IP, al que se la han eliminado una serie de componentes físicos y funcionalidades lógicas que no necesita para su trabajo, mientras que se le han añadido otros componentes de hardware y de software que le ayudan en su trabajo de enrutamiento.

Como todo ordenador, un router necesita un sistema de arranque (bootstrap), encargado de realizar un chequeo del resto de los componentes antes de pasar el control a un sistema operativo (Cisco IOS, en el caso de los routers Cisco).

El sistema de arranque se almacena en una memoria ROM (Read Only Memory = Memoria de Solo Lectura), junto con una parte básica del sistema operativo, la que toma el control inicialmente, mientras que el cuerpo principal de éste se almacena en una memoria especial, de tipo FLASH, que se puede borrar y reprogramar, permitiendo con ello las actualizaciones necesarias. El contenido de la memoria Flash se conserva en caso de cortes de energía o durante los reinicios del router.

Por otra parte, las funcionalidades operativas de los routers son configurables mediante una serie de instrucciones escritas en un fichero de texto, denominado archivo de configuración, que se almacena en un módulo de memoria de tipo NVRAM (No Volatil RAM), cuyo contenido se conserva durante un corte de energía o si se reinicia el equipo.

Una vez inicializado un router, el fichero de configuración es cargado en una memoria RAM (Random Access Memory=Memoria de Acceso Aleatorio), desde la que se va ejecutando el conjunto de órdenes en él contenido. También se almacenan en esta memoria las tablas de enrutamiento, encargadas de almacenar los puertos del router por los que son accesibles las diferentes máquinas.

Por último, el router posee una serie de puertos o interfaces físicas, puntos de conexión del mismo con las diferentes redes a las que está unido, y a través de los cuales se produce la entrada y salida de datos al equipo. El número de interfaces depende del tipo y funcionalidades del router (y de su precio, claro).

14.3 Tipos de Router.

Los tipos de router a usar en una red varían dependiendo del tipo de ésta, del número de usuarios y de la función o funciones que deba desempeñar, pudiendo variar mucho la complejidad y el precio de ellos en función del tipo elegido.

Si queremos segmentar nuestra red en diferentes subredes, nos hará falta un router de segmentación, con tantos puertos Ethernet como subredes queremos crear (más los de enlace con otros routers), siendo

siempre conveniente que nos sobren puertos, con vista a futuras ampliaciones en la red. Cada subred utilizará luego un hub concentrador o un switch para dar acceso a sus clientes individuales.

Podemos desear un ancho de banda dedicado para un número elevado de equipos individuales, prescindiendo así de los hubs. Necesitaremos entonces un routers de concentración, que precisa aún más puertos Ethernet, aunque no suele ser necesario que sean de alta velocidad de transmisión.

Para conectar una red corporativa a Internet necesitaremos un router de frontera, que actuará como gateway de la red interna, recogiendo todos aquellos paquetes de datos destinados a máquinas externas.

En caso de tener que conectar dos redes WAN o dos segmentos de red en sucursales o campus diferentes, necesitaremos un routers de backbone, que proporciona transporte óptimo entre nodos de la red, con interfaces de alta velocidad que proporcionan un elevado ancho de banda. Generalmente estarán basados en tecnología de fibra óptica.

Por último, también es posible al acceso a redes inalámbrico a redes mediante routers con tecnología wireless, un medio práctico de liberar los equipos de las limitaciones de los cables físicos.

14.4. Protocolos de Enrutamiento.

Hemos visto antes que los routers mantienen unas tablas de enrutamiento, en las que van anotando las direcciones IP de las máquinas destino y los puertos adecuados para darles salida de forma óptima.

Los routers suelen encontrarse interconectados entre ellos, pasándose los paquetes de datos de uno a otro, hasta llegar a la máquina destino. Como cada router tan solo es responsable de las máquinas directamente conectadas a él (incluyendo los routers vecinos), se hace necesario un mecanismo que permita a los routers comunicarse entre sí, para evitar que cada uno tenga en sus tablas registros inválidos.

Esto se consigue por medio de una serie de protocolos de enrutamiento, responsables de que los diferentes routers mantengan sus tablas de enrutamiento acordes, obteniéndose una red convergente. Con ello se consigue, por ejemplo, que si un ordenador o un servidor se apaga en una red, los routers sepan que ya no está accesible, evitando el envío de datos que no llegarán a su destino, y disminuyendo con ello el tráfico de red.

Para mantener las tablas de enrutamiento actualizadas, un router pueden mandar a los routers vecinos una copia de su tabla cada determinado periodo de tiempo (enrutamientos por vector de distancia) y también cuando alguna máquina en su red sufre algún cambio (enrutamiento por estado de enlace). Depende del protocolo de enrutamiento con que funcione.

Existen diferentes protocolos de comunicación entre routers, cada uno de los cuales utiliza mecanismos propios para conseguir la convergencia en la red y para determinar el mejor camino que puede seguir un paquete de datos en su viaje hasta la máquina destino, y cada uno utiliza un sistema de determinación de mejor ruta (métrica) diferente.

Según su misión en una red podemos diferenciar dos tipos principales de protocolos de enrutamiento: los protocolos de gateway interior (IGP), encargados de la comunicación entre routers de una misma red, entre los que destacan RIP e IGRP, y los protocolos de gateway exterior (EGP) o de frontera, encargados de la comunicación entre routers de redes diferentes.

Entre los más importantes protocolos de enrutamiento podemos destacar los siguientes:

RIP (Protocolo de Información de Enrutamiento), es un protocolo de enrutamiento por vector de distancia que calcula las distancias hacia la máquina destino en función de cuántos routers debe atravesar un paquete para llegar a su destino (saltos), enviando cada paquete de datos por el camino que en cada momento muestre una menor distancia. RIP actualiza las tablas de enrutamiento a intervalos programables, generalmente cada 30 segundos. Es un buen protocolo de enrutamiento, pero necesita que constantemente se conecten los routers vecinos, generándose con ello una gran cantidad de tráfico de red.

IGRP (Protocolo de Enrutamiento de Gateway Interior), desarrollado por Cisco System, es un protocolo de enrutamiento por vector de distancia que usa una métrica compuesta basada en diferentes variables de red, como ancho de banda, unidades máximas de transmisión (MTU), confiabilidad, etc. Envía actualizaciones de las tablas de enrutamiento cada 90 segundos.
EIGRP (Protocolo de Enrutamiento de Gateway Interior Mejorado), protocolo mixto basado en IGRP, basado en una métrica de vector distancia, pero que manda actualizaciones de las entradas de las tablas que han cambiado por haber sido alterado el estado de alguna máquina de su red.

OSPF, protocolo puro de estado de enlace, que calcula las rutas más cortas y accesibles mediante la construcción de un mapa de la red y el mantenimiento unas bases de datos con información sobre su sistema local y sobre los vecinos. Cuando una máquina de su sistema cambia, se envía esa entrada de la tabla a los routers vecinos.

El protocolo de enrutamiento a elegir en cada caso depende del tipo de red (LAN, WAN, etc.), de su topología y del uso de la misma, siendo posible en la mayoría de los casos configurar varios protocolos en un mismo router.

14.5 Seguridades de Redes.

La mejor manera para tener un mejor control en el nivel de seguridad en nuestro Router Linux, es la siguiente:

- La ubicación del equipo debe situarse fuera del alcance de personal ajeno del que administra la red de la empresa.
- La PC en la que se instale el Router Linux debe prestar un excelente rendimiento de todo el equipo, por lo tanto debe ser armado con piezas de calidad y de buena marca para no tener problemas en la instalación de interfaces.
- En lo referente a la seguridad dentro de la red se tiene que precautelar que la administración del equipo se la haga localmente, o desde una dirección IP autorizada para acceder al equipo.
- Una de la nuevas formar de bloquear el acceso prohibido al router es instalar un cortafuegos, ya sea un equipo independiente que limite el acceso a la red o el mismo PC que nos está sirviendo de router instalar un Firewall, una de las nuevas formas de implementar un Firewall es mediante iptables.

  IPtables es un sistema de firewall vinculado al kernel de linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo. Al igual que el anterior sistema ipchains, un firewall de iptables no es como un servidor que lo iniciamos o detenemos o que se pueda caer por un error de programación (esto es una pequeña mentira, ha tenido alguna vulnerabilidad que permite DOS, pero nunca tendrá tanto peligro como las aplicaciones que escuchan en determinado puerto TCP): iptables esta integrado con el kernel, es parte del sistema operativo. ¿Cómo se pone en marcha? Realmente lo que se hace es aplicar reglas. Para ellos se ejecuta el comando iptables, con el que añadimos, borramos, o creamos reglas. Por ello un firewall de iptables no es sino un simple script de shell en el que se van ejecutando las reglas de firewall.

Linux

15.1 Instalación

Una instalación de tipo personalizada le permitirá una mayor flexibilidad en el proceso de instalación. Podrá elegir su esquema de particionamiento, los paquetes que desea instalar y mucho más. La instalación de tipo personalizado es más apropiada para usuarios a los que les son familiares las instalaciones de Red Hat Linux y para aquéllos que temen perder flexibilidad.

Los requisitos de espacio recomendado en disco para una instalación personalizada son los siguientes:

Personalizada (mínimo): 475 MB

Personalizada (escogiendo todo): 5.0 GB

Como se puede deducir del nombre, una instalación personalizada hace énfasis en la flexibilidad. Tiene control completo sobre los paquetes que serán instalados en su sistema.

15.2 Arranque de Linux

La pantalla de Bienvenida no le pedirá ninguna información. Por favor lea el texto de ayuda en el panel de la izquierda para instrucciones adicionales e información sobre el registro de su producto Red Hat Linux.

Observe que el botón Esconder ayuda se encuentra en la parte inferior izquierda de la pantalla. La pantalla de ayuda aparece abierta por defecto. Si no quiere visualizar la información, haga click en Esconder ayuda para minimizar esta parte de la pantalla.

Haga click en Siguiente para continuar.

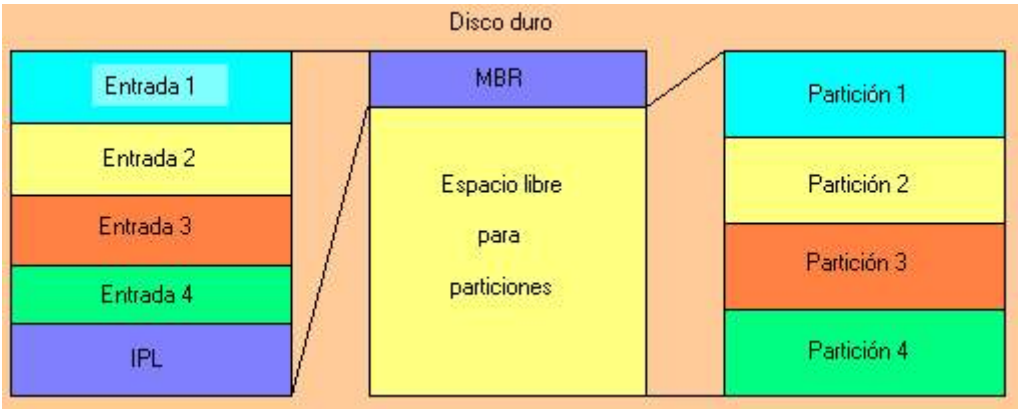
La selección del idioma apropiado le ayudará también a seguir los pasos correctos para configurar la zona horaria (huso horario), más tarde. El programa de instalación intentará definir el huso horario adecuado basándose en su configuración.

Elija qué tipo de instalación desea realizar (consulte la Figura 3-11). El sistema Red Hat Linux le permitirá elegir el tipo de instalación que mejor se ajuste a sus necesidades. Las opciones disponibles son: Estación de trabajo, Servidor, Portátil, Personalizada y Actualización.

15.3 Dispositivos y particiones en Linux

Los discos duros se pueden dividir, de manera estándar, en hasta cuatro partes independientes de tamaño arbitrario (llamadas particiones), para tener por ejemplo distintos sistemas operativos (OS).

La estructura general de los discos duros es la siguiente:



Como vemos, lo primero que hay es lo que se llama el MBR (Master Boot Record), también llamado tabla de partición, y después viene el espacio de disco que pueden usar libremente las particiones. Por supuesto, no tiene que haber exactamente cuatro particiones hechas, podemos tener sólo una, dos, tres o las cuatro, según nuestras necesidades.

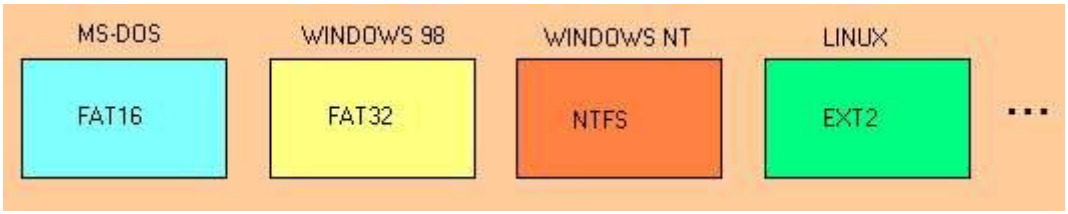
El MBR es el lugar donde se indican cuántas particiones hay y cómo son. Para esto hay cuatro entradas, donde se guarda la información relativa a cada posible partición: si existe o no, de qué tipo es, en qué posición se encuentra dentro del disco duro, cuánto ocupa, flag de activo y demás.

Además, el MBR contiene un pequeño programa (IPL), que podemos llamar gestor de arranque principal, encargado de arrancar una partición u otra. Este programita se ejecuta al arrancar el ordenador, después de los chequeos y otras cosas que hace la Bios.

15.4. Creación de las particiones en Linux

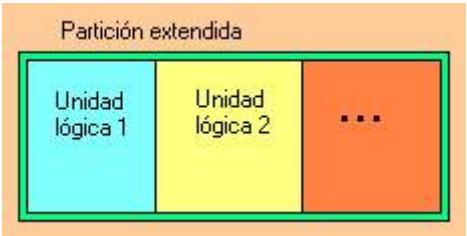
De forma más general, las particiones pueden ser primarias o extendidas. La diferencia entre unas y otras está en su estructura interna.

Hay muchos tipos de particiones primarias, dependiendo del sistema de archivos que utilice la partición. Así por ejemplo, el Ms-Dos usa particiones tipo FAT16, el Windows 95/98 las FAT32, Windows NT/2000 las NTFS, Linux las EXT2, etc.



Hemos de tener en cuenta que el tipo FAT16 tiene un tamaño máximo de 2G. Por eso, aunque Windows 95/98 se puede instalar en una partición FAT16, si queremos usar una partición de más de 2G, ha de usarse FAT32.

Las particiones extendidas son unas particiones especiales que a su vez están divididas internamente en partes independientes (llamadas unidades lógicas), pero el número de unidades lógicas no está limitado. Las unidades lógicas son parecidas a las particiones primarias, pero están dentro de particiones extendidas. Una partición extendida no es útil por sí misma, su única función es ser el recipiente de unidades lógicas.



Hay hasta 256 tipos de particiones y cada una tiene un identificador numérico. Para conocerlos, se pueden usar varios programas, por ejemplo el fdisk de Linux. He aquí algunos:

0x05 Partición extendida DOS
0x06 Partición FAT16
0x07 Partición NTFS
0x0B Partición FAT32

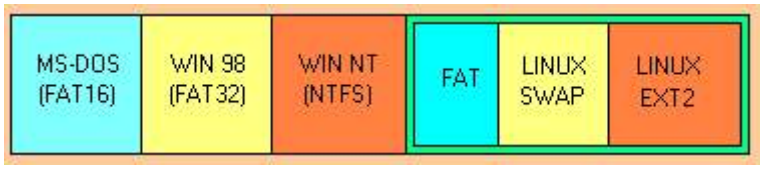
0x16 Partición FAT16 oculta
0x17 Partición NTFS oculta
0x1B Partición FAT32 oculta

0x82 Partición Linux Swap
0x83 Partición Linux Ext2

Todos los OS se pueden instalar en particiones primarias, pero sólo algunos en unidades lógicas. Por ejemplo, Linux se puede instalar en unidades lógicas mientras que los OS de Microsoft no. Éstos últimos requieren obligatoriamente particiones primarias. Usan las unidades lógicas sólo para guardar datos. Esto

limita el número de OS de este tipo que se pueden instalar, dado que como mucho puede haber hasta cuatro particiones primarias.

El hecho de que no esté limitado el número de unidades lógicas es una característica muy interesante para Linux. Todas las particiones de Linux, incluidas las de Swap, se pueden instalar en unidades lógicas. No es obligatorio usar particiones primarias. De esta manera, instalando Linux en unidades lógicas aún podemos tener hasta tres OS más, en sus correspondientes particiones primarias. Por ejemplo, podemos tener un disco duro como el siguiente:



15.5 Creación del espacio de intercambio (swap)

El tamaño de la partición swap viene determinado por la cantidad de memoria RAM en su sistema y la cantidad de espacio disponible en su disco duro. Por ejemplo, si tiene 128 MB de RAM la partición swap creada puede ser 128 MB - 256 MB (dos veces su RAM), dependiendo de cuanto espacio de disco haya disponible.

Una partición de 100 MB (montada como /boot) en la que residen el kernel de Linux y los archivos relacionados.

Una partición raíz montada como / en la que están almacenados todos los archivos (el tamaño exacto de su partición depende del espacio de disco disponible).

15.6 Creación de los sistemas de ficheros (ext-3)

Con la versión Red Hat Linux 7.2, Red Hat cambia el sistema de ficheros por defecto con el formato ext2 al sistema de ficheros journaling ext3.
Características de ext3

Básicamente, el sistema de ficheros ext3 es una versión mejorada de ext2. Las mejoras introducidas proporcionan las siguientes ventajas:

Disponibilidad

Tras un corte eléctrico o una caída inesperada del sistema (también se denomina cierre no limpio del sistema), se debe comprobar con el programa e2fsck cada sistema de ficheros ext2 montado en la máquina para ver si es consistente. El proceso de comprobación lleva mucho tiempo y puede prolongar el tiempo de arranque del sistema de un modo significativo, especialmente si hay grandes volúmenes que contienen un elevado número de ficheros. Durante este proceso, no se puede acceder a los datos de los volúmenes.

Con la característica journaling del sistema de ficheros ext3 ya no es necesario realizar este tipo de comprobación en el sistema de ficheros después de un cierre no limpio del sistema. En el sistema ext3, únicamente se realiza una comprobación de consistencia en los casos puntuales en los que se producen determinados errores de hardware, como, por ejemplo, fallos en el disco duro. El tiempo empleado para recuperar un sistema de ficheros ext3 tras un cierre no limpio del sistema no depende del tamaño del sistema de ficheros ni del número de ficheros, sino del tamaño del "journal" (diario) utilizado para mantener la consistencia en el sistema. Por defecto, la recuperación del tamaño del "journal" tarda alrededor de un segundo, según la velocidad del hardware.
Integridad de los datos

El sistema de ficheros ext3 proporciona una integridad superior de los datos si se produce un cierre no limpio del sistema. El sistema de ficheros ext3 le permite seleccionar el tipo y el nivel de protección de los datos. Por defecto, Red Hat Linux 7.3 configura los volúmenes ext3 para que el nivel de consistencia de los datos sea elevado en relación con el estado del sistema de ficheros.
Velocidad

El sistema de ficheros ext3, aparte de permitir escribir datos más de una vez, en la mayoría de los casos tiene un rendimiento superior al que proporciona ext2 porque los "journals" de ext3 optimizan el movimiento de los cabezales de los discos duros. Se pueden seleccionar tres modos de journaling para optimizar la velocidad, pero, como contrapartida, la integridad de los datos se verá afectada.

Fácil transición

La migración de ext2 a ext3 es muy sencilla y se pueden aprovechar las ventajas de un sólido sistema de ficheros con journaling sin tener que volver a dar formato al sistema.

Si realiza una instalación nueva de Red Hat Linux 7.3, el sistema de ficheros por defecto que se asigna a las particiones Linux del sistema es ext3. Si realiza una actualización a partir de una versión de Red Hat Linux con particiones ext2, el programa de instalación le permitirá convertir estas particiones a ext3 sin perder los datos. Consulte el apéndice Actualización del sistema actual de Manual oficial de instalación de Red Hat Linux para obtener más información.

En las siguientes secciones se describirán los pasos para crear y configurar las particiones ext3. Si tiene particiones ext2 y está ejecutando Red Hat Linux 7.3, puede omitir las secciones en las que se explican los pasos para particionar y dar formato al disco, y, en su lugar, puede ir directamente a la la sección de nombre Conversión al sistema de ficheros ext3.

Si está añadiendo un nuevo disco duro al sistema Red Hat Linux y desea utilizar el sistema de ficheros ext3, primero debe particionar el disco duro con un programa, como, por ejemplo, fdisk, y después dar formato al sistema de ficheros.
Particionamiento con fdisk

Para usar fdisk, abra un indicador de comandos de shell y conéctese como usuario raíz. El comando fdisk precisa que se especifique el dispositivo que se está particionando como argumento en el comando. En los siguientes ejemplos, el dispositivo será /dev/hdb, que corresponde al segundo dispositivo del canal IDE primario. Para comenzar, escriba:

/sbin/fdisk /dev/hdb

En la siguiente tabla se proporcionan los comandos fdisk más comunes.

Tabla 5-1. Comandos fdisk

Comando	Función
m	muestra la ayuda
p	muestra la tabla de particiones actual
d	borra una partición
n	crea una partición nueva
w	escribe la tabla de particiones en el disco
t	establece el tipo de sistema de ficheros anticipado para la partición
l	muestra la lista de tipos de sistemas de ficheros para particiones
q	sale de fdisk sin aplicar cambios en el disco

Sugerencia

Si necesita salir del programa en cualquier momento sin aplicar los cambios en el disco, escriba q.

Una vez en el programa fdisk, escriba n para crear una nueva partición. El programa le solicitará que elija un tipo de partición: seleccione e para una partición extendida y p para una partición primaria.

Antes de seleccionar el tipo de partición, recuerde que el sistema Red Hat Linux sólo admite cuatro particiones primarias en un disco. Si desea crear más particiones, puede configurar una (solamente una) de las cuatro particiones primarias como extendida para que actúe como contenedor de una o más particiones lógicas. La partición extendida, puesto que actúa como contenedor, debe tener como mínimo

un tamaño igual al tamaño total de todas las particiones lógicas que contiene. Para obtener más información sobre particiones de disco, consulte el apéndice Introducción a particiones de discos en Manual oficial de instalación de Red Hat Linux.

Después de seleccionar el tipo de partición y el número de esa partición, elija el cabezal del cilindro en el que desearía que comience la partición. Puede escribir [Intro] para excluir el valor por defecto.

A continuación, especifique el tamaño. El modo más sencillo de realizar esto es escribir + tamaño M, donde tamaño es el tamaño de la partición en megabytes. Si presiona [Intro] sin introducir un valor, fdisk utilizará el resto del disco.

Repita este proceso hasta que haya creado el esquema de particionamiento deseado.

15.7 Redes con TCP/IP

TCP/IP es un conjunto de protocolos de comunicaciones desarrollado para permitir a un conjunto de computadoras cooperar y compartir recursos a través de una red de comunicaciones. De entre sus muchas características, hay dos que lo han transformado en uno de los protocolos de mayor difusión:
Es un estándar abierto, diseñado independientemente de plataformas de hardware y software específicas. Así, TCP/IP es ideal para interconectar sistemas mas allá de lo diferentes que éstos sean.
Es independiente de la tecnología física que se utilice para construir la infraestructura de la red. TCP/IP puede montarse sobre Ethernet, Token-Ring, enlaces seriales telefónicos (dial-up links), redes X.25, y virtualmente cualquier otro medio físico de transmisión de datos.

Arquitectura de TCP/IP

La familia TCP/IP está formada por múltiples protocolos de diferentes propósitos. Algunos de ellos, tales como IP, TCP y UDP constituyen el mecanismo básico de transmisión de datos, y serán utilizados por todas las aplicaciones. Otros protocolos permiten realizar tareas mucho más específicas, tales como transferir archivos entre computadoras (FTP), obtener páginas o documentos de la Web (HTTP), o sincronizar la hora desde otro equipo (XNTP).

Cualquier aplicación real utilizará varios de esos protocolos. Un caso típico es el envío de correo electrónico. En primer lugar, existe un protocolo para enviar y recibir correo electrónico (denominado SMTP), que define una serie de comandos que una máquina envía a la otra cuando requiere transferirle un mensaje. Esos comandos permiten especificar quien es el autor del mensaje, a quien va dirigido y cual es el texto a enviar. Sin embargo, ese protocolo (como todos los otros protocolos de aplicación) asume que hay alguna manera confiable para comunicar datos entre ambas computadoras, limitándose simplemente a definir a muy alto nivel los comandos necesarios para manejar la transmisión, pero no los detalles acerca de como va a efectivizarse la misma. Dichos detalles son dejados en manos de alguno de los protocolos de menor nivel, llamados protocolo de transporte: TCP o UDP.

SMTP utiliza a TCP como protocolo de transporte. TCP es responsable de asegurar que los comandos trasmitidos lleguen al otro extremo de la comunicación, contabilizando qué ha sido transmitido ya y retransmitiendo toda información que no haya llegado exitosamente a destino. Si la información a transmitir es demasiado larga, TCP la segmentará en varios paquetes que se transmitirán individualmente.

Obsérvese que esta funcionalidad se requiere para muchas aplicaciones; es por ello que conforma un protocolo independiente en vez de formar parte de la especificación de protocolos como SMTP. Desde el punto de vista del programador, TCP es una libraría de rutinas que las aplicaciones utilizan cuando necesitan comunicaciones confiables con otra computadora a través de la red.

De manera similar, TCP utiliza los servicios de IP para efectivamente desplazar los paquetes alrededor de la red. IP constituye un protocolo de red, y es el encargado de determinar que rutas deberán seguir los paquetes para llegar al punto de destino desde el punto de origen. Nuevamente, IP se presenta como una librería que utilizan protocolos de transporte como TCP al momento de enviar la información.

Esta estrategia para construir protocolos en varios niveles se denomina diseño estratificado (o layering). Consiste en considerar a las aplicaciones, a TCP y a IP como diferentes capas, cada una de las cuales hace

uso de los servicios ofrecidos por la capa inmediatamente inferior. En general, la arquitectura de las aplicaciones basadas en TCP/IP presentan 4 capas:

- Un protocolo de aplicación, para tareas específicas (por ejemplo, correo electrónico).
- Un protocolo de transporte, que provee servicios de extremo a extremo (como TCP o UDP).
- El protocolo de red IP, que provee el encaminamiento de los paquetes a su destino final.
- Un protocolo de enlace físico, que provee acceso al medio físico de transmisión (por ejemplo, Ethernet o X.25).

15.8 Configuración de TCP/IP

En un momento dado podrían existir, entre las computadoras de origen y destino, múltiples conexiones ocurriendo simultáneamente; pensemos, por ejemplo, en varios usuarios abriendo sesiones remotas o transfiriendo simultáneamente archivos o correo electrónico entre dos máquinas de la red.

Claramente, no es suficiente lograr que los paquetes lleguen al destino correcto; es necesario además poder discriminar a cual conexión pertenecen de las múltiples conexiones simultáneas que pueden existir en un momento dado.

Para identificar cada conexión, TCP asigna un número de puerto a cada una. Supongamos que tres personas están transfiriendo archivos entre dos computadoras. TCP asignaría un número de puerto a cada transferencia, por ejemplo, 1000, 1001 y 1002. Todos los paquetes que se envíen como parte de una misma conexión tendrán asignado ese número como puerto de origen. El número de puerto de origen permite también establecer una correspondencia directa entre una conexión de red y el programa de usuario que interviene en uno de los extremos de la conversación. En el otro extremo, habrá otro programa que recibe los datos transmitidos, que también deberá poder asociarse a dicha conexión. Esa asociación se hace por medio del puerto de destino que TCP asigna a cada paquete que transmite.

Cuando un programa de usuario (conocido como proceso cliente) abre una conexión de red por medio de TCP, se le asigna (mas o menos al azar) un número de puerto. Ese programa asume que en la otra computadora estará en ejecución otro programa (conocido como proceso servidor o, en la jerga Unix, demonio de red) que espera recibir peticiones desde la red. Cuando ese programa fue iniciado, su capa TCP le asignó también un número de puerto. Obviamente, el número de puerto que se asigne a procesos servidores no puede ser aleatorio, ya que sería imposible para los clientes saber que número especificar como puerto de destino. Los procesos servidores se asocian, entonces, con números de puerto fijos (llamados "números bien conocidos" -- "well-known numbers"), mientras que los procesos cliente obtienen números de puertos aleatorios al iniciar las conexiones12.

Obsérvese que una conexión de red puede entonces identificarse unívocamente por medio de un conjunto de 4 números: las direcciones IP de ambos extremos y los números de puerto de origen y destino. Para el caso de las tres transferencias de archivos que se ponían como ejemplo mas arriba, si las direcciones IP de las maquinas de origen y destino son 172.16.10.150 y 172.16.8.123, y la transferencia se hace utilizando el protocolo FTP (que tiene asignado el número de puerto 21), cada conexión se puede identificar de la siguiente manera:

Conexión 1
Dirección IP: 172.16.10.150
Puerto: 1000 Dirección IP: 172.16.8.123
Puerto: 21

Conexión 2
Dirección IP: 172.16.10.150
Puerto: 1001 Dirección IP: 172.16.8.123
Puerto: 21

Conexión 3
Dirección IP: 172.16.10.150
Puerto: 1002 Dirección IP: 172.16.8.123

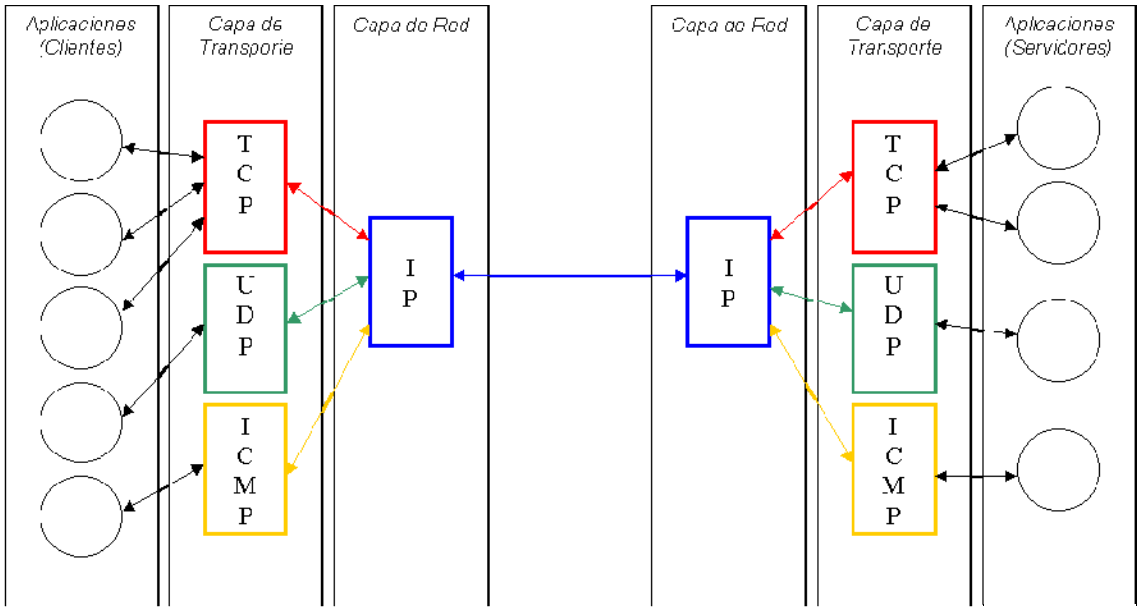
Puerto: 21

No pueden existir dos conexiones que compartan el mismo conjunto de números, pero es suficiente con que al menos uno sea diferente. En el ejemplo anterior, en donde tres usuarios transfieren archivos entre dos computadoras, dado que las computadoras involucradas en cada transferencia son las mismas, las direcciones IP son iguales para cada conexión y todos realizan transferencias vía FTP, por lo que el puerto de destino para las tres conexiones es el 21. Lo único que difiere es el número de puerto de origen, que permite diferenciar a los tres usuarios.

Cada par formado por una dirección IP y un número de puerto se denomina socket (enchufe), por lo que una conexión TCP puede verse como un canal virtual a través de una red, "enchufada" a un socket en cada extremo. Por otra parte, el utilizar un único canal de comunicaciones para combinar múltiples conexiones de datos se denomina multiplexación; la información que arriba desde la red debe ser demultiplexada a fin de que cada módulo de software reciba los paquetes que le corresponden.

De hecho, hay varios niveles de multiplexación en TCP/IP. Por una parte, TCP la utiliza para mantener múltiples conexiones, tal como se describió previamente. Por otra parte, sin embargo, existen otros protocolos (como UDP e ICMP) que utilizan IP como un medio para distribuir paquetes a lo largo de la red. Cuando IP recibe paquetes entrantes desde la red, debe poder determinar a cual protocolo de mayor nivel pasar el paquete. Esto constituye también otra forma de demultiplexación, y se realiza por medio de la asignación a cada paquete, por parte del IP de origen, de un numero de protocolo. Dicho número tiene un rol similar al número de puerto, con la diferencia de que no identifica conexión sino el protocolo de transporte que está administrando esa conexión.

El proceso de multiplexación y demultiplexación de TCP/IP se esquematiza en la siguiente figura:



En resumen...

Una red TCP/IP está formada por múltiples redes interconectadas por medio de gateways. Dichos gateways pueden ser dispositivos físicos especializados (llamados routers) o bien computadoras con múltiples adaptadores de red (llamados multihomed hosts).

Cada una de esas redes estará formada por máquinas individuales (los hosts de la red) o por subredes interconectadas. Cada máquina de la red recibirá un identificador numérico único, llamado dirección IP.

Las computadoras de la red ejecutarán aplicaciones que establecerán comunicaciones entre ellas por medio de protocolos como TCP ó UDP, los cuales utilizarán el protocolo IP para rutear paquetes de información entre el origen y el destino.

Algunas computadoras de la red ofrecerán servicios a las demás, estableciéndose relaciones de tipo cliente/servidor entre ellas. Los roles de cliente y servidor no son excluyentes; una misma maquina puede al mismo tiempo ser cliente y servidor. Mas aun, podría ocurrir que la relación cliente/servidor se dé entre dos procesos ejecutándose en la misma máquina.
Los procesos servidores reciben peticiones desde la red, usualmente "escuchando" en puertos fijos de TCP (llamados números bien conocidos). Los clientes, por otra parte, utilizan puertos TCP asignados mas o menos al azar al iniciar la conexión.
El par formado por una dirección IP y un número de puerto se denomina socket. Una conexión puede identificarse unívocamente por el par de sockets correspondientes al nodo de origen y al de destino.

15.9 Configuración de SLIP

El protocolo SLIP (Serial Line Internet ProtocolProtocolo de línea serie) permite utilizar TCP/IP mediante una línea serie, como puede ser una línea telefónica con módem o cualquier tipo de línea dedicada asíncrona. Por supuesto, para usar SLIP tiene que tener acceso a un servidor SLIP. Muchas empresas y universidades proporcionan acceso SLIP por poco dinero.

Existen principalmente dos programas relacionados con SLIP: dip y slattach. Ambos se usan para iniciar una conexión SLIP sobre un dispositivo serie. Es necesario utilizar uno de estos dos programas para habilitar SLIP; no es suficiente con llamar al servidor SLIP con programas como kermit y después usar las instrucciones ifconfig y route. Esto se debe a que dip y slattach realizan una llamada del sistema ioctl() especial para hacerse con el control del dispositivo serie y ponerlo a disposición de la interfaz de SLIP.

Con dip puede llamarse a un servidor SLIP, hacer ciertas negociaciones de entrada con el mismo (intercambio de usuario y password, por ejemplo) y después iniciar la conexión SLIP. Por su lado, slattach se limita prácticamente a modificar la línea serie para que la utilice SLIP, por lo que está indicado para líneas dedicadas que no requieren interacción con el módem o ni ninguna negociación de protocolo. Casi todo el mundo, sin embargo, usa dip.

Con dip también puede configurar su sistema como servidor SLIP, permitiendo a otras máquinas hacer una llamada telefónica y conectarse a la red a través de su módem y su conexión Ethernet. Vea la documentación y los manuales en línea de dip para más información.

A SLIP se le llama conexión ``punto a punto'' pues a ambos lados de la línea existen sólo las dos máquinas involucradas (no como sucede en una ethernet). Esta idea se generaliza y mejora con el protocolo PPP (point-to-point protocol).

Cuando inicia una conexión al servidor SLIP, se le asignará una dirección IP, bien de forma ``estática'' (su dirección IP es siempre la misma) o ``dinámica'' (su dirección puede ser diferente de un día para otro). Por lo general, los valores de la dirección y pasarela asignados serán impresos por el servidor SLIP al conectarse. El programa dip es capaz de capturar esos valores y configurar su sistema para adaptarse a ellos.

Esencialmente, configurar una conexión SLIP es como configurar la conexión en ``bucle local'' o con Ethernet. En las siguientes líneas le mostramos las diferencias. Es importante que vea lo que hemos explicado antes sobre configuración en general de TCP/IP, y aplique ahora las modificaciones que le vamos a contar.

Si tiene una línea dedicada o un cable conectado directamente al servidor SLIP, no es necesario usar dip para iniciar la conexión. En su lugar puede usar slattach para configurar el dispositivo SLIP.

En este caso, el fichero /etc/rc.inet1 puede quedar como sigue:

Slattach asigna el primer dispositivo SLIP disponible (sl0,sl1, etc.) a la línea serie especificada.

Observe que el primer parámetro de slattach es el protocolo SLIP a utilizar. Actualmente solo valen slip y cslip. Slip es el SLIP normal, y cslip es un SLIP que incluye compresión de las cabeceras de los datagramas. Por ello su elección habitual será cslip a menos que tenga algún problema con esta conexión.

Si tiene más de una interfaz SLIP tendrá que tener en cuenta algunas cosas respecto al rutado. Tiene que decidir qué rutas añadir, y esto debe hacerse en función de la configuración de la red a la que se conecte.

Le serán de ayuda los libros sobre configuración de TCP/IP, la documentación en línea de la orden route, etc.

15.10 Configuraciones adicionales

Routers Linux con Zebra

Zebra es un software que permite montar routers sobre sistemas operativos tipo Unix. Este software dispone de una interfaz de configuración basada en el Cisco IOS, por lo que será útil a los administradores familiarizados con routers Cisco.

La instalación desde un RPM es sencilla:

rpm -iUvh zebra-0.92a-3.i386.rpm

Quedando todo instalado.

Zebra instala cinco daemons que escuchan en puertos consecutivos. A continuación una tabla muestra cuales son los daemons y en que puertos escuchan:

zebra	2601 tcp
ripd	2602 tcp
ripngd	2603 tcp
ospfd	2604 tcp
bgpd	2605 tcp

Utilice la forma que usted considere adecuada para iniciar los daemons en su sistema, estos estan ubicados en /usr/local/sbin para instalaciones desde el código fuente.

En instalaciones RPM puede arrancar los daemons de la siguiente manera:

/etc/rc-d/init.d/zebra start
/etc/rc-d/init.d/ripd start
/etc/rc-d/init.d/ripngd start
/etc/rc-d/init.d/ospfd start
/etc/rc-d/init.d/bgpd start

Los archivos de configuración estan en /usr/local/etc/. Con las instalación de las fuentes viene un archivo de ejemplo para cada uno de los daemons. Estos no son utilizados directamente por Zebra, hay que cambiarles el nombre. Son los siguientes:

Nombre original	Fichero de configuración
zebra.conf.sample	zebra.conf
ripd.conf.sample	ripd.conf
ripngd.conf.sample	ripngd.conf
ospf.conf.sample	ospf.conf
bgpd.conf.sample	bgpd.conf

En el caso de haber realizado una instalación a partir de un RPM no es necesario que modifique los archivos de configuración. De todas formas puede localizarlos en /etc/zebra

Para realizar la configuración del router puede acceder directamente a cada uno de los protocolos (daemons) que utiliza zebra. Simplemente haga un telnet al puerto que desee, por ejemplo, para configurar rip:

$ telnet localhost ripd

De todas formas esto no es necesario, puesto que Zebra proporciona una herramienta que integra todos los protocolos/daemons. Esta herramienta es vtysh:

```
$ vtysh

Hello, this is Zebra (version 0.92a).
Copyright 1996-2001 Kunihiro Ishiguro

zebra>
```

Si ha configurado alguna vez un router Cisco esto le resultará familiar.

Accedemos al modo enable y empezamos con la configuración:

```
zebra>en
zebra#sh run
Building configuration
Current configuration
!!
zebra#
```

No hay configuración, por lo que reiniciaremos el demonio zebra para que vuleva a leerla, por ejemplo:

```
/etc/rc.d/init.d/zebra restart
```

Entramos de nuevo en zebra:

```
$ vtysh

Hello, this is Zebra (version 0.92a).
Copyright 1996-2001 Kunihiro Ishiguro
mypc>
mypc>en
mypc#sh run
Building configuration
Current configuration
!!i
nterace lo
!i
nterface eth0
!
mypc#
```

A partir de aquí la configuración es similar a la de un router Cisco.

Conclusiones y Recomendaciones

16.1 Conclusiones.

- Se pudo demostrar que es posible montar un equipo de ruteo IP con software libre, lo cual implica una reducción de costos significativo para la implementación de un sistema de red en una PYMES.

- Se logro construir un ruteador basado en un servidor, esto nos permite además de contar con un router, tenemos la posibilidad de montar un firewall, servidor DNS, servidor DHCP y una de las alternativas más interesantes hacer un mantenimiento desde la web del equipo que funciona como servidor.

- Linux a demostrado ser un esfuerzo mancomunado por tratar de mejorar la fiabilidad de los SO, además existen estándares de updates y upgrades para lograr un desarrollo sostenido del sistema, no van ha existir problemas de incompatibilidad entre versiones del SO Linux, permitiéndonos tener un desarrollo continuo y estandarizado.

- Tenemos que indicar que además de conocer sobre las funciones de ruteo, se crea una dependencia del SO, pero esto se compensa al tener un software libre y muy estable.

16.2 Recomendaciones.

- El equipo en el que se va ha montar el servidor linux se recomienda que tenga piezas probadas y de marcas conocidas para no tener problemas con incompatibilidades de los dispositivos en el momento de instalar el SO Linux.

- Se tiene que tener un conocimiento medio sobre la administración de un sistema operativo Linux, para no tener problemas con las configuraciones y el mantenimiento del sistema.

- Se tiene que tener seguridades en el equipo en el cual se instala el servidor Linux, se tiene que trabajar sobre cuentas de usuarios y no sobre el Súper usuario, se debe instalar los paquetes estrictamente necesarios para el funcionamiento del equipo, por el motivo que se pueden instalar agujeros de seguridad debido a que el momento de instalar los paquetes se configura con las opciones de default y esto puede traer vulnerabilidades al sistema.