



DIPLOMADO EN LINUX

NIVEL 3. SERVIDORES EN REDES LINUX

**CREAD JOSÉ ACEVEDO Y GÓMEZ
Gloria Esther Ricardo, Luis Hernando Rojas, Rogelio Vásquez
BOGOTÁ, D.C. 2002**

SERVIDORES EN REDES LINUX

1. INTRODUCCIÓN

Dado el avance tecnológico, el crecimiento de las organizaciones y la necesidad de comunicación de todos los seres humanos, nacen las redes de comunicación.

Estas redes hacen que todos estemos comunicándonos, compartiendo información y recursos de diferente tipo. Es por ello que a través del presente capítulo daremos a conocer diferentes conceptos sobre redes, sus ventajas y elementos sofisticados que permiten que la información viaje y sea segura.

Dado que se está trabajando en un sistema operativo específico como *Linux* el cual requiere una plataforma robusta y segura, se tratarán aspectos relevantes sobre los servidores que requiere este sistema.

Esperamos con nuestro aporte apoyar el proceso de formación de la comunidad universitaria y todas aquellas personas que puedan tener interés en el tema.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

El propósito de este nivel es dar a conocer conceptos básicos necesarios sobre redes de comunicación y los diversos elementos que las componen.

2.1.1 Objetivos específicos

En este nivel el lector conocerá o recordará algunos conceptos que requiere para poder avanzar tanto en el nivel tres del curso como en el nivel cuatro. Mediante este nivel el alumno:

Conocerá conceptos generales de redes.

Verá los requerimientos mínimos para que exista una red.

Conocerá la importancia dentro de las comunicaciones del modelo de referencia OSI.

Obtendrá una descripción de los diferentes medios físicos de transmisión.

Analizará las diferentes topologías de red existentes.

Interpretará como los protocolos son indispensables en el campo de las comunicaciones.

Conocerá e interpretará el concepto del modelo Cliente/Servidor.

Obtendrá información sobre los diversos servidores existentes y verá la necesidad de profundizar en el tema para lograr hacer instalación de los servidores requeridos por el sistema operacional Linux .

3. REDES

3.1 CONCEPTOS GENERALES DE REDES

A continuación mencionaremos algunos conceptos que son importantes para el entendimiento del tema de las redes.

3.1.1 Red: Básicamente una red es un conjunto de elementos conectados entre sí con el fin de prestar un servicio. Pero como debemos hablar de términos técnicos decimos que una red se define como un conjunto de nodos que son capaces de comunicarse entre sí, a menudo contando con los servicios de nodos dedicados que comparten datos entre los participantes.

3.1.2 Clasificación de las redes: Las redes según su cubrimiento geográfico se pueden clasificar como redes área local (*Lan*) y redes de amplia cobertura (*Wan*).

3.1.2.1 Redes de Área local: Son aquellas que conectan una red de ordenadores normalmente confinadas en un área geográfica, como un solo edificio o un área cercana. Sin embargo, no son necesariamente simples de planificar, ya que pueden unir muchos centenares de ordenadores y pueden ser usadas por miles de usuarios. El desarrollo de varias normas de protocolos de red (normas para entendimiento) y medios físicos han hecho posible la proliferación de este tipo de redes en todo tipo de organizaciones.

3.1.2.2 Redes de Amplia Cobertura: Conectan múltiples redes *Lan* que están geográficamente dispersas. Esto se realiza conectando las diferentes *lan* mediante servicios que incluyen líneas telefónicas alquiladas (punto a punto), líneas de teléfono normales y enlaces vía satélite entre otros.

3.1.3 Ventajas de las redes: Teniendo en cuenta el concepto anterior es necesario mencionar algunas ventajas de las redes *lan*, tales como:

- ✓ Procesamiento distribuido: Significa que cada estación de trabajo en una red ejecuta sus propias aplicaciones. Esto permite que un gran número de estaciones se

incorpore a una red sin reducir el poder de procesamiento disponible para las estaciones individuales.

- ✓ Comunicaciones rápidas: Es una función de las redes que asegura que la información que se accesa sea reciente y actualizada.
- ✓ Recursos compartidos: Es una función de ahorro de las redes. Los usuarios pueden tener acceso a periféricos (impresoras, *modems*, *gateways*, dispositivos gráficos, etc) lo que elimina la necesidad de comprar equipo innecesario.
- ✓ Optimización de recursos existentes: Las organizaciones pueden utilizar su hardware existente y sacar un mayor provecho del software de red.

3.2 Componentes Básicos de una Red: Para que exista red es necesarios que existan unos componentes mínimos, de los cuales mencionaremos los siguientes:

- ✓ Servidor: Es el encargado de suministrar un servicio a solicitud de un cliente.
- ✓ Cliente: Es aquel que solicita un servicio
- ✓ Nodos: Son los puntos de la red que han de interconectarse.
- ✓ Nic (Tarjeta de Interfase de Red): Es una tarjeta de interfase que permite a un nodo comunicarse con otros nodos o servidores de archivos.
- ✓ Cableado: Es el medió físico que hace posible la comunicación.
- ✓ Sistema Operativo de red: Es aquel que controla el uso de la red. Deben proporcionar los medios para transferir bits de una estación a otra. Para permitir que las estaciones de trabajo aprovechen la conectividad y que operen en colaboración, se requieren funciones de software de alto nivel. Estas funciones son los servicios de red. Los servicios de red más comunes son: archivo, impresión, almacenamiento masivo, comunicaciones, correo, bases de datos.

3.2.1 Enlazando redes: Dado el crecimiento de las redes, los usuarios ya no están interesados en establecer una Lan individual solamente. Por el contrario están buscando la posibilidad de conectar sus Lans individuales a redes más amplias. La conexión entre redes se refiere a la interconexión de una Lan a otra o a una red privada o pública. Hay varios dispositivos disponibles para conectar las redes:

- ✓ Enrutadores (*Routers*): Son dispositivos inteligentes de conexión que pueden enviar paquetes al segmento correcto de la red y así llevarlos a su destino. Proporcionan la capacidad de dirigir diferentes tipos de protocolos a varios puertos.
- ✓ Pasarelas (*Gateways*): Proporcionan los medios para conectar sistemas diferentes como los procesadores centrales o minicomputadoras a Lans. Un *gateway* hace más que simplemente leer una dirección de paquete, en realidad reestructura el paquete. De esta forma, la puerta es un traductor de protocolo.
- ✓ Repetidores: Son la forma más simple de conectar dos *Lans*. Estos dispositivos amplifican y vuelven a formar las señales en una *Lan* y las pasan a otra. El problema con los repetidores es que no filtran el tráfico, sino que solamente la amplifican.
- ✓ Puentes (*Bridges*): Los puentes conectan dos o más *Lans*. Un puente esta formado por hardware y software que hace de dos Lans una red más grande. Los puentes normalmente conectan a las redes con protocolos y arquitectura comunes.

3.2.2 Modelo de referencia OSI: Es un modelo para la arquitectura de red, la cual procura hacer posible la comunicación entre computadoras diferentes en ambientes diversos. El modelo OSI consiste en una jerarquía de siete niveles que define las características eléctricas, los estándares de comunicación y las aplicaciones de software para los sistemas de computo.

3.2.2.1 NIVEL FISICO: Este nivel incluye el tipo de cable, la potencia de la señal y las distancias. Sus características principales son:

- ✓ Transmisión de bits por un canal
- ✓ Garantiza la llegada de cada bit sin error
- ✓ Manejas los niveles de voltaje, duración de bit y el establecimiento de liberación de conexiones.

3.2.2.2 NIVEL DE ENLACE: Este nivel determina la estrategia y mecanismos para tener acceso al cable, la forma que tomará la información transmitida y como se reagrupara en su destino. Sus características principales son:

- ✓ Conexión libre de error
- ✓ Manejo de tramas
- ✓ Soluciona tramas dañadas, perdidas o duplicadas.

3.2.2.3 NIVEL DE RED: Este nivel determina la ruta de la información de un nodo al siguiente. Sus características principales son:

- ✓ Ofrece independencia de las tecnologías de transmisión utilizadas en la red.
- ✓ Enruta los paquetes a través de la red.

3.2.2.4 NIVEL DE TRANSPORTE: Este nivel especifica como manejar los errores y la retransmisión de los datos. Sus características principales son:

- ✓ Asegurar una transmisión fidedigna entre nodos a través de la red.
- ✓ Control de extremo a extremo.
- ✓ Manejo de secuencia, duplicados, errores, pérdidas.

3.2.2.5 NIVEL DE SESION: Este nivel indica lo que se llaman los niveles superiores del modelo OSI. Estos niveles hacen que el software establezca conectividad entre nodos. Esto agrega una dimensión lógica a la conectividad física proporcionada por los niveles inferiores. Este proceso lógico controla la conexión y desconexión de la trayectoria de comunicación de datos (la sesión), y establece las reglas bajo las cuales los nodos tendrán conversación. Sus características principales son:

- ✓ Provee estructura de control para la comunicación.
- ✓ Establece, administra y finaliza sesiones entre dos aplicaciones.

3.2.2.6 NIVEL DE PRESENTACION: Una vez hecha la conexión física a través de los cinco niveles inferiores, es necesario determinar el formato de la información. Este nivel especifica las convenciones de código y datos para los programas de aplicación. Su característica principal es:

- ✓ Transformación de datos (compresión de la información y la codificación).

3.2.2.7 NIVEL DE APLICACIÓN: Este nivel especifica los protocolos de servicio de archivos que se usan directamente en una aplicación. Proporciona un formato estándar para que los usuarios finales puedan enviar y recibir correo electrónico, leer y actualizar archivos remotos o acceder y cambiar registros de la base de datos.

En resumen los niveles 1 y 2 del modelo OSI se refieren al *hardware* de la LAN incluyendo el cable, el hardware que accesa el cable y el software que controla el hardware. Los niveles 3 y 4 se conocen con frecuencia como los protocolos de comunicación. Los niveles 5, 6 y 7 se consideran como las funciones del sistema operativo que proporcionan servicios a las aplicaciones.

3.2.3 Medios físicos de transmisión: Entre los medios que permite la conexión de las redes se encuentran los siguientes:

3.2.3.1 Satélite: Es un sistema de comunicación que utiliza un satélite terrestre en órbita como un punto medio para lograr la reflexión de las ondas electromagnéticas generadas por una estación transmisora y otra receptora, ambas ubicadas en puntos distantes.

Microondas: La información se transmite por un rayo de luz entre dos estaciones terrestres.

3.2.3.2 Cableado: existen diferentes tipos de cables, cada uno con características específicas que permiten una mejor transmisión de la información y una mayor seguridad:

- ✓ Par trenzado: El cable de par trenzado no apantallado, o UTP, ofrece muchas ventajas respecto de los cables coaxiales, dado que los coaxiales son ligeramente caros y requieren algún tipo de cuidado durante la instalación. El cable UTP es similar, o incluso el mismo, al cable telefónico que puede estar instalado y disponible para la red en muchos edificios.
- ✓ Coaxial: Esta compuesto por un alambre conductor protegido por un blindaje entrelazado que actúa como tierra. Existe el coaxial grueso y el fino. El primero de ellos se empleaba generalmente, para crear grandes troncales. Una troncal una pequeños segmentos de una red LAN, es excelente porque puede soportar muchos nodos en una topología de bus y el segmento puede ser muy largo. Un segmento de cable coaxial grueso puede tener hasta 500 metros de longitud y máximo 100 nodos conectados.

Cable coaxial fino: También conocido como *Ethernet* 10 base-2, ofrece muchas de las ventajas de la topología de bus del coaxial grueso, con un costo menor y una instalación más sencilla. Es considerablemente más delgado y más flexible, pero sólo puede soportar 30 nodos, cada uno separado por un mínimo de 0.5 metros.

- ✓ Fibra óptica: Es una fibra muy fina hecha en dos tipos de vidrio, una para el hilo central y otra para el exterior. Es indispensable para situaciones donde las emisiones electrónicas y los riesgos medioambientales varían. La norma Ethernet permite segmentos de cable de fibra óptica de dos kilómetros de longitud, haciendo *Ethernet* a fibra óptica perfecto para conectar nodos y edificios que de otro modo no podrían ser conectados con cable de cobre.

3.2.4 Topologías de red: La topología se refiere a la forma como están dispuestas físicamente las estaciones en una red. Son tres las topologías básicas, lo demás son combinaciones que se hacen de estas y se conoce como híbridas. Las topologías son:

- ✓ Bus: En esta topología, las estaciones están dispuestas a lo largo de un solo cable que se puede extender hasta uno de los extremos (bus de comunicaciones). Un árbol es un bus lineal complejo en donde el cable se divide en uno o en ambos de sus extremos, pero que ofrece solamente una trayectoria de transmisión entre cualesquiera de las dos estaciones.

Ethernet es una red en bus la cual requiere una cantidad mínima de cables y tiene soporte de muchos adaptadores para la red. Permite un buen equilibrio entre velocidad, costo y facilidad de instalación.

- ✓ **Anillo:** En una topología de anillo, las estaciones están dispuestas a lo largo de la trayectoria de transmisión de manera que la señal pasa a través de una estación a la vez antes de regresar a la estación de origen. Las estaciones forman un círculo o anillo.
- ✓ **Estrella:** En una topología en estrella, hay un nodo central que se conecta a cada estación a través de un simple enlace cliente a cliente. Cualquier comunicación entre una estación a otra debe pasar a través del nodo central.

3.3 PROTOCOLOS DE COMUNICACIÓN

Los protocolos de comunicación son estándares de *software* o *hardware* que controlan la transmisión entre dos estaciones. En computadores personales los programas de comunicaciones ofrecen una variedad de protocolos como: *kermit*, *xmodem*, *zmodem*, etc, para transferir archivos mediante los modem. En redes *LAN* los protocolos están incluidos en *Ethernet*, *Token Ring* y otros métodos de acceso. En redes de mainframe existen múltiples niveles de protocolos y protocolos dentro de los protocolos. El protocolo es una empresa compleja que administra las redes de grandes organizaciones .

Los protocolos están presentes en todas las etapas necesarias para establecer una comunicación entre equipos de cómputo, desde aquellas de más bajo nivel (e.g. la transmisión de flujos de bits a un medio físico) hasta aquellas de más alto nivel (e.g. el compartir o transferir información desde una computadora a otra en la red). Tomando al modelo OSI (*Open Systems Interconnection*) como referencia podemos afirmar que para cada capa o nivel que él define existen uno o más protocolos interactuando. Los protocolos son entre pares (*peer-to-peer*), es decir, un protocolo de algún nivel dialoga con el protocolo del mismo nivel en la computadora remota.

3.3.1 Protocolo TCP/IP

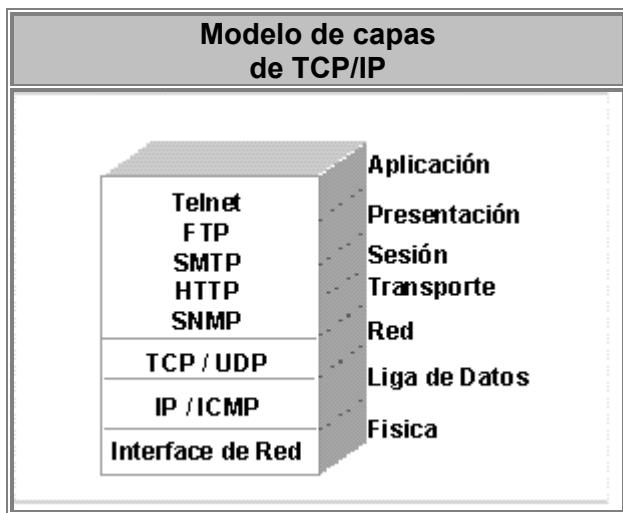
En la actualidad, las funciones propias de una red de computadoras pueden ser divididas en las siete capas propuestas por ISO para su modelo de sistemas abiertos (OSI).

Conjunto de Protocolos TCP/IP Su relación con el Modelo OSI						
Aplicación						
Presentación	TELNET	FTP	SNMP	SMTP	DNS	HTTP
Sesión						
Transporte	TCP					
Red	IP					
Liga de Datos	802.2				X.25	LLC/SNAP
	802.3	802.5		LAPB		ATM
Física	Ethernet	Token Ring	FDDI	Línea Síncrona WAN		SONET

- **TCP** = *TRANSFER CONTROL PROTOCOL*
- **IP** = *INTERNET PROTOCOL*

Sin embargo la implantación real de una arquitectura puede diferir de este modelo. Las arquitecturas basadas en TCP/IP proponen cuatro capas en las que las funciones de las capas de Sesión y Presentación son responsabilidad de la capa de Aplicación y las capas de Liga de Datos y Física son vistas como la capa de Interface a la Red. Por tal motivo para TCP/IP sólo existen las capas Interface de Red, la de Intercomunicación en Red, la de Transporte y la de Aplicación. Como puede verse TCP/IP presupone independencia del medio físico de comunicación, sin embargo existen estándares bien definidos a los nivel de Liga de Datos y Físico que proveen mecanismos de acceso a los diferentes

medios y que en el modelo TCP/IP deben considerarse la capa de Interface de Red; siendo los más usuales el proyecto *IEEE802, Ethernet, Token Ring y FDI*



Descripción del Modelo de Capas de TCP/IP

Capa de Aplicación.	Invoca programas que acceden servicios en la red. Interactúan con uno o más protocolos de transporte para enviar o recibir datos, en forma de mensajes o bien en forma de flujos de bytes.
Capa de Transporte.	Provee comunicación extremo a extremo desde un programa de aplicación a otro. Regula el flujo de información. Puede proveer un transporte confiable asegurándose que los datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente de tal manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota, esto lo hace añadiendo identificadores de cada una de las aplicaciones. Realiza además una verificación por suma, para asegurar que la información no sufrió alteraciones durante su transmisión.
Capa Internet.	Controla la comunicación entre un equipo y otro, decide qué rutas deben seguir los paquetes de información para alcanzar su destino. Conformar los paquetes IP que será enviados por la capa inferior. Desencapsula los

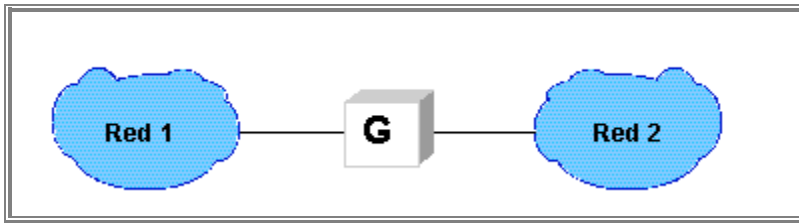
	paquetes recibidos pasando a la capa superior la información dirigida a una aplicación.
Capa de Interface de Red.	Emite al medio físico los flujos de bit y recibe los que de él provienen. Consiste en los manejadores de los dispositivos que se conectan al medio de transmisión.

3.3.2 Arquitectura de interconexión de redes en TCP/IP

Para entender el funcionamiento de los protocolos TCP/IP debe tenerse en cuenta la arquitectura que ellos proponen para comunicar redes. Tal arquitectura ve como iguales a todas las redes a conectarse, sin tomar en cuenta el tamaño de ellas, ya sean locales o de cobertura amplia. Define que todas las redes que intercambiarán información deben estar conectadas a una misma computadora o equipo de procesamiento (dotados con dispositivos de comunicación); a tales computadoras se les denomina compuertas, pudiendo recibir otros nombres como enrutadores o puentes.

Arquitectura de Interconexión de Redes en TCP/IP Características
<ul style="list-style-type: none"> • Protocolos de no conexión en el nivel de red. • Conmutación de paquetes entre nodos. • Protocolos de transporte con funciones de seguridad. • Conjunto común de programas de aplicación.

Arquitectura de Interconexión de Redes en TCP/IP Interconexión de Redes
<ul style="list-style-type: none"> • Las redes se comunican mediante compuertas. • Todas las redes son vistas como iguales.



Direcciones IP	
<ul style="list-style-type: none"> • Longitud de 32 bits. • Identifica a las redes y a los nodos conectados a ellas. • Especifica la conexión entre redes. • Se representan mediante cuatro octetos, escritos en formato decimal, separados por puntos. 	

Para que en una red dos computadoras puedan comunicarse entre sí ellas deben estar identificadas con precisión Este identificador puede estar definido en niveles bajos (identificador físico) o en niveles altos (identificador lógico) dependiendo del protocolo utilizado. TCP/IP utiliza un identificador denominado dirección internet o dirección IP, cuya longitud es de 32 bits. La dirección IP identifica tanto a la red a la que pertenece una computadora como a ella misma dentro de dicha red.

Clases de Direcciones IP

Clases	Número de Redes	Número de Nodos	Rango de Direcciones IP
A	127	16,777,215	1.0.0.0 a la 127.0.0.0
B	4095	65,535	128.0.0.0 a la 191.255.0.0
C	2,097,151	255	192.0.0.0 a la 223.255.255.0

		Bits de la dirección IP									
Clase	0	1	2	3	4	8	16	24	31		
A	0	id. de red					id. de nodo				
B	1	0	id. de red				id. de nodo				
C	1	1	0	id. de red			id. de nodo				
D	1	1	1	0	dirección multiemisión						
E	1	1	1	1	0	reservado para usos futuros					

Tomando tal cual está definida una dirección IP podría surgir la duda de cómo identificar qué parte de la dirección identifica a la red y qué parte al nodo en dicha red. Lo anterior se resuelve mediante la definición de las "Clases de Direcciones IP". Para clarificar lo anterior veamos que una red con dirección clase A queda precisamente definida con el primer octeto de la dirección, la clase B con los dos primeros y la C con los tres primeros octetos. Los octetos restantes definen los nodos en la red específica.

Se ha mencionado que el enrutamiento sirve para alcanzar redes distantes. También se señaló que las direcciones IP se agrupan en clases. Ahora bien para cada clase se pueden contar con un número determinados de subredes. Las subredes son redes físicas independientes que comparten la misma dirección IP (es decir aquella que identifica a la red principal). La pregunta entonces es cómo se logra que equipos que comparten el mismo identificador de red pero se sitúan en redes físicas diferentes podrán comunicarse usando compuertas? La solución a este problema es determinando una máscara de dirección.

Recordemos que los protocolos TCP/IP están enfocados a la transmisión de paquetes de información, buscando la independencia de la arquitectura de la red. Arquitecturas como la Ethernet logran la comunicación sólo mediante el conocimiento de la dirección física de las computadoras. Así en cada computadora que opere con el protocolo IP debe contar con algún procedimiento para la translación de la dirección IP a la dirección física de la computadora con la que establezca comunicación.

Una conversión dinámica de direcciones Internet a direcciones físicas es la más adecuada, debido a que se obtiene la dirección física por respuesta directa del nodo que posee la dirección IP destino. Una vez que la dirección física se obtiene ésta es guardada en una tabla temporal para subsecuentes transmisiones, de no ser así podría haber una sobrecarga de tráfico en la red debido a la conversión de direcciones por cada vez que se transmitiera un paquete.

3.3.3 Protocolo Internet (IP)

El Protocolo Internet proporciona un servicio de distribución de paquetes de información orientado a no conexión de manera no fiable. La orientación a no conexión significa que los paquetes de información, que será emitido a la red, son tratados independientemente, pudiendo viajar por diferentes trayectorias para llegar a su destino. El término no fiable significa más que nada que no se garantiza la recepción del paquete.

**Protocolo Internet (IP)
Características**

- Protocolo orientado a no conexión.
- Fragmenta paquetes si es necesario.
- Direccionamiento mediante direcciones lógicas IP de 32 bits.
- Si un paquete no es recibido, este permanecerá en la red durante un tiempo finito.
- Realiza el "mejor esfuerzo" para la distribución de paquetes.
- Tamaño máximo del paquete de 65635 bytes.
- Sólo se realiza verificación por suma al encabezado del paquete, no a los datos que este contiene.

La unidad de transferencia máxima determina la longitud máxima, en *bytes*, que podrá tener un datagrama para ser transmitida por una red física. Obsérvese que este parámetro está determinado por la arquitectura de la red: para una red *Ethernet* el valor de la MTU es de 1500 *bytes*. Dependiendo de la tecnología de la red los valores de la MTU pueden ir desde 128 hasta unos cuantos miles de *bytes*.

3.3.4 Fragmentación

La arquitectura de interconexión de redes propuesta por TCP/IP indica que éstas deben ser conectadas mediante una compuerta. Sin obligar a que la tecnología de las redes físicas que se conecten sea homogénea. Por tal motivo si para interconectar dos redes se utilizan medios con diferente MTU, los datagramas deberán ser fragmentados para que puedan ser transmitidos. Una vez que los paquetes han alcanzado la red extrema los datagramas deberán ser reensamblados.

Existen dos tipos de enrutamiento; el directo y el indirecto. Debido a que en el enrutamiento directo los datagramas se transmiten de un equipo a otro, en la misma red física, el proceso es muy eficiente. La vinculación entre la dirección física y la IP se realiza mediante el ARP. En el indirecto la transmisión del datagrama se efectúa mediante la

intercesión de las compuertas. Aquí la compuerta que actúa como enrutador debe de estar provista de mecanismos para conocer, y por tanto decidir, la trayectoria de la red que se desea alcanzar.

Enrutamiento Indirecto

En este direccionamiento un equipo debe enviar a una compuerta el datagrama con destino a una red física distante. La compuerta de la red física envía el datagrama a otras compuertas hasta alcanzar a aquel que puede emitirlo en forma directa a la red destino. La compuerta debe conocer las rutas hacia las diferentes redes externas, ellas pueden utilizar a su vez un enrutamiento indirecto en el caso de no conocer la ruta a una red específica. Las compuertas conocen las trayectorias a otra red mediante Tablas de Enrutamiento.

Tablas de Ruteo IP

Este es el algoritmo comúnmente utilizado para el enrutamiento de IP. Las tablas de enrutamiento están presentes en todo equipo que almacene información de cómo alcanzar posibles destinos. En las tablas no se almacena la ruta específica a un equipo, sino aquella a la red donde se encuentre. Cada puerto de comunicación de la compuerta debe poseer una dirección IP.

Rutas por *Default*

- Si cada tabla de ruteo conservara información sobre todos los

destinos posibles, el espacio sería insuficiente.

- Es necesario que con un mínimo de información, el equipo pueda tomar decisiones de ruteo.
- Una técnica para mantener tablas de ruteo pequeñas consiste en enviar los datagramas a destinos predeterminados (redes predeterminadas).

Como se vió en la arquitectura de interconexión de redes de TCP/IP cada par de redes se conectan mediante compuertas. Para que los paquetes alcancen sus redes destino las compuertas deben contar con mecanismos mediante los cuales intercambien la información de las redes que conecta cada uno. En la arquitectura de enrutamiento por compuerta núcleo existe una compuerta que centraliza las funciones de enrutamiento entre redes, a esta compuerta se le denomina núcleo. Cada compuerta en las redes a conectar tiene como compuerta por default a la compuerta núcleo. Varias compuertas núcleo pueden conectarse para formar una gran red; entre las compuertas núcleo se intercambiará información concerniente a las redes que cada una de ellas alcanzan. La arquitectura centralizada de enrutamiento fue la primera que existió. Sus principales problemas radican no tanto en la arquitectura en sí, si no en la forma en que se propagaban las rutas entre las compuertas núcleo.

3.3.5 Protocolo de control de transferencia

Proporciona un mecanismo fiable para la transferencia de flujos de información. Aunque está íntimamente relacionado con IP TCP es un protocolo independiente de propósito general. Al ser un protocolo de alto nivel su función es que grandes volúmenes de información lleguen a su destino correctamente, pudiendo recobrar la pérdida esporádica de paquetes.

**Protocolo de Control de
Transferencia**

- Proporciona comunicación bidireccional completa mediante circuitos virtuales.
- Desde el punto de vista del usuario la información es transmitida por flujos de datos.
- Confiabilidad en la transmisión de datos por medio de:
 - Asignación de números de secuencia a la información segmentada.
 - Validaciones por suma.
 - Reconocimiento de paquetes recibidos.
 - Utiliza el principio de ventana deslizante para esperar reconocimientos y reenviar información.

A cada paquete que es enviado se le asigna un número de identificador, el equipo que lo recibe deberá enviar un reconocimiento de dicho paquete, lo que indicará que fue recibido. Si después de un tiempo dado el reconocimiento no ha sido recibido el paquete se volverá a enviar. Obsérvese que puede darse el caso en el que el reconocimiento sea el que se pierda, en este caso se reenviará un paquete repetido.

Fiabilidad en la transferencia de TCP

- Cada vez que un paquete es enviado se inicializa un contador de tiempo, al alcanzar el tiempo de expiración, sin haber recibido el reconocimiento, el paquete se reenvía.
- Al llegar el reconocimiento el tiempo de expiración se cancela.

El concepto de la Ventana Deslizante

- Se define un tamaño de la ventana, que serían el número de paquetes a enviar sin esperar reconocimiento de ellos.
- Conforme se recibe el reconocimiento de los primeros paquetes transmitidos la ventana avanza de posición enviando los paquetes siguientes.
- Los reconocimientos pueden recibirse en forma desordenada.

Si el protocolo sólo contara con reconocimientos positivos gran parte de la capacidad de la red estaría desperdiciada, pues no se enviarían más paquetes hasta recibir el reconocimiento del último paquete enviado. El concepto de ventana deslizante hace que exista una continua transmisión de información, mejorando el desempeño de la red.

Este protocolo deja al programa de aplicación a ser explotado la responsabilidad de una transmisión fiable. Con él puede darse el caso de que los paquetes se pierdan o bien no sean reconstruidos en forma adecuada. Permite un intercambio de datagramas más directo entre aplicaciones y puede elegirse para aquellas que no demanden una gran cantidad de datagramas para operar óptimamente.

3.3.6 Computación cliente/servidor

El término cliente/servidor describe un sistema en el que una máquina cliente solicita a una segunda máquina llamada servidor que ejecute una tarea específica.

El cliente suele ser una computadora personal común conectada a una *LAN*, y el servidor es, por lo general, una máquina anfitriona, como un servidor de archivos PC, un servidor de archivos de *Unix*, o un servidor de archivos de *Linux* . Las peticiones de trabajo pueden incluir distintas tareas, como por ejemplo:

- ✓ Regreso de todos los registros de la base de datos de archivos de clientes donde el nombre del cliente = Pedro.
- ✓ Almacenar este archivo en un directorio de datos específico del servidor de archivos.
- ✓ Conectarse a un centro de computo determinado y recuperar ciertos elementos.

- ✓ Subir este paquete de datos a la macrocomputadora de la empresa.

Se considera que la tecnología cliente servidor es el camino a seguir debido a que las organizaciones deben afrontar el reto de mantener sus negocios al día y ser competitivos. Los sistemas cliente/servidor distribuyen las áreas de actividad importantes de la empresa en varias unidades distintas y, aunque están integradas entre sí, se pueden considerar independientes.

Una solución cliente/servidor tiene varias ventajas sobre la red centralizada tradicional. Las tareas se dividen entre el cliente y el servidor para conseguir un funcionamiento de red más eficiente.

3.3.6.1 Áreas de aplicación de cliente/servidor:

- ✓ Archivos compartidos
- ✓ Procesos especializados
- ✓ Acceso a bases de datos
- ✓ Interfaces hombre máquina.

La aplicación cliente/servidor más común es un sistema de administración de bases de datos que usa SQL, la consulta a la base de datos se envía desde el cliente pero se procesa en el servidor, solo los resultados se envían por la red al cliente.

La seguridad es importante por cuanto los datos se encuentran en un servidor o un número limitado de servidores, en los cuales se encuentran restricciones de acceso dependiendo del tipo de cliente a ingresa.

3.3.6.2 Cómo construir un buen primer sistema cliente servidor: En primer lugar, lo más importante es considerar qué secciones de la empresa se van a desarrollar utilizando este modelo. Para ello es necesario tener en cuenta beneficios como la reducción de costos y por ende una mayor utilidad del negocio, la reutilización de equipos o el compartir ciertos elementos que estén subutilizados.

En segundo lugar, no se debe diseñar una aplicación vital o crucial en este modelo hasta que no se haya demostrado que este concepto de cliente/servidor funcionará en su organización.

Por último, una estrategia sería escoger una aplicación que se encuentre en medio de las dos anteriores. Un sistema de cliente/servidor que produzca beneficios sin impactar las áreas críticas de la empresa, dará la oportunidad de capacitar al personal que utiliza estos sistemas, tiempo para valorar el impacto en las LAN, y la experiencia para desarrollar sistemas mayores y mejores, recibiendo así mayores y mejores utilidades de la inversión.

3.3.6.3 Concepto de servidores: Los sistemas operativos de red deben proporcionar no sólo los medios de transferir bits de una estación de trabajo en una Lan a otra. Para permitir que las estaciones aprovechen esta conectividad y que operen en colaboración, se requieren funciones de software de alto nivel. Estas funciones son los servicios de red. El software de servicios de red corre en la plataforma del servidor, abasteciendo funcionalidad en la red. Los servidores proporcionan al usuarios desde el servicio de red más básico, como bloqueo de registros y archivos, hasta servicios más complejos como compartir la base de datos del servidor.

3.3.7.4 Tipos de servidores:

- ✓ Servidores de archivos: El objeto de transferir un archivo es mover un archivo o parte de él de donde se encuentre (archivo fuente) a cualquier otra parte (archivo destino). Normalmente el servicio de transferencia de archivos se usa interactivamente por un usuario en línea o también puede ser invocado por una aplicación. Un servidor de archivos contiene software que forma una protección alrededor del sistema operativo de discos normal de la computadora. Esta protección filtra los comandos hacia el servidor de archivos antes de que el sistema operativo pueda recibirlos.

- ✓ Servidor de impresión: Permiten al usuario compartir la estación de trabajo y las impresoras de red. Los servicios de control de listas de espera y notificación de alertas hacen que el servicio de impresión sea más fácil de manejar. Un servidor de impresión de red puede ser una micro computadora dedicada que sólo ejecuta el software del servidor de impresión, o puede ser una sección de software que se ejecute en el servidor de archivos de la red.

- ✓ Servidor de mensajería (correo): Proporcionan la capacidad de almacenamiento y envío a una red. Estos servicios son una captación, distribución y envío de mensajes y datos a sistemas similares y distintos. Esto permite que el correo electrónico y otras aplicaciones puedan enviar información a través de redes locales y de área extendida. Se han establecido diferentes estándares de mensajería como:
 - ✓ X.400 es el estándar de mensajería para OSI.
 - ✓ SMTP (Protocolo de transferencia de correo simple), para redes TCP/IP.
 - ✓ Sistema de manipulación de mensajes (MHS), utilizado en *netware*.
 - ✓ El Sendmail de sistema operativo *Linux* .

En el siguiente tema se trataran los servidores específicos del sistema operativo Linux tales como:

- ✓ Servidor DNS(Servidor de nombres de dominio): Este es utilizado para resolver nombres de nodos. Este servidor se ha creado para que haga la asociación entre una palabra y un número. Así no hace falta, saber cuál es la máquina cuyo número es 192.168.234.123 pero sabemos que la máquina que esta al lado del café se denomina grajo, por ejemplo.
- ✓ Servidor NIS(Sistema de Información de redes): Proporciona facilidades de acceso genérico a las bases de datos que puede ser usado para distribuir información como la contenida en los ficheros *passwd* y *groups* entre los nodos de la red. Esto hace que la red aparezca como un sistema único, con las mismas cuentas en todos los nodos.
- ✓ Servidor UUCP(Servidor de Protocolos): Tiene como objetivo primario enviar archivos conteniendo noticias de la *Usenet* vía línea marcada. Aunque es posible también utilizar correo electrónico vía UUCP, lo cual exige que el servidor dé algunos permisos adicionales al usuario.
- ✓ Servidor Samba: Es un conjunto de programas que interpreta el protocolo SMB que utilizan las máquinas Windows para comunicarse. Se usa en la configuración de una red que necesita ofrecer o acceder a máquinas e impresoras Windows.

3.4 Redes con TCP/IP

Linux soporta una implementación completa de los protocolos de red TCP/IP (*Transport Control Protocol/Internet Protocol*). TCP/IP ha resultado ser hasta ahora el mejor mecanismo de comunicación entre ordenadores de todo el mundo. Con Linux y una tarjeta *Ethernet* podrá introducir su máquina en una red local o (si se tienen las conexiones apropiadas) a la Internet, la red TCP/IP de ámbito mundial. Poner en marcha una pequeña red local de máquinas *Unix* es fácil. Sólo requiere una tarjeta Ethernet en cada máquina y los cables adecuados así como hardware accesorio (terminadores, etc). Y si su universidad o empresa tiene acceso a la Internet, podrá insertar su máquina Linux en esta red. La implementación actual de TCP/IP y los protocolos relacionados para Linux se llama "NET2". No tiene que ver con la versión NET-2 para BSD. En realidad, se refiere a que es la segunda implementación que se hace para Linux .

NET-2 de *Linux* soporta también SLIP (*Serial Line Internet Protocol*). SLIP le permitirá acceder a la Internet con un módem. Si su universidad o empresa proporciona accesos por SLIP, podrá llamar desde su casa al servidor SLIP y conectarse así a la Red. Recíprocamente, si posee en *Linux* una tarjeta de Red y un *modem* podrá configurar un servidor SLIP en él.

Para obtener más información de configuración de TCP/IP en Linux , le animamos a que lea el documento *NET-2 HOWTO*, disponible mediante FTP anónimo en *sunsite.unc.edu*. Se trata de una guía completa de configuración, que incluye conexiones mediante *Ethernet* y SLIP. Otro documento relacionado es el *Ethernet HOWTO*, que se centra en cómo configurar diversos modelos de tarjetas *Ethernet*.

Además, en el Proyecto de Documentación de Linux, al que pertenece esta guía, se ha desarrollado otro sobre este tema, *Linux Network Administrator's Guide*.

Encontrará más ayuda en el libro *TCP/IP Network Administration*, de *Craig Hunt*. Contiene información completa acerca del uso y la configuración de TCP/IP en máquinas Unix.

3.4.1 Hardware requerido

Puede utilizar el TCP/IP para Linux sin hardware de red. Así podrá usar el dispositivo "loopback" para conectarse con usted mismo. Aunque parezca poco serio, hay algunos programas que necesitan conexiones de red "loopback" para funcionar. Sin embargo, si quiere usar Linux en una red TCP/IP *Ethernet*, necesitará una de las tarjetas soportadas: *3com 3c503, 3c503/16; Novell NE1000, NE2000; Western Digital WD8003, WD8013; Hewlett Packard HP27245, HP27247, HP27250*. Se ha comprobado que también funcionan las siguientes tarjetas clónicas: Clónicas de WD-80x3: LANNET LEC-45; clónicas de NE2000: Alta Combo, Artisoft LANtastic AE-2, Asante Etherpak 2001/2003, D-Link Ethernet II, LTC E-NET/16 P/N 8300-200-002, Network Solutions HE-203, SVEC 4 Dimension Ethernet, 4-Dimension FD0490 EtherBoard 16, D-Link DE-600 y SMC Elite 16. Sobre este tema encontrará más información en el documento *Ethernet HOWTO*.

Linux también funciona con SLIP, que permite acceder a la red Internet por teléfono. En este caso, necesitará un módem compatible con el servidor SLIP. Muchos servidores requieren módems de alta velocidad, a 14400 bits por segundo (norma V.32bis).

3.4.2 Configuración de TCP/IP

En esta sección intentaremos explicar cómo configurar una conexión TCP/IP con *Ethernet*. Nótese que este método funcionará en muchos sistemas, pero no siempre. Nuestra explicación debería ser suficiente para aclararle el camino en la configuración de red en su máquina, pero hay además otros detalles que no mencionaremos aquí por su extensión. Le aconsejamos que consulte los documentos *Linux Network Administrators' Guide* y *NET-2 HOWTO* para más información.

En primer lugar, vamos a asumir que su sistema Linux ha sido instalado con el software TCP/IP. Esto incluye clientes como telnet y ftp, comandos de administración como ifconfig y route (que suelen estar en /etc) y ficheros de configuración de red, como /etc/hosts. Los documentos adicionales que hemos mencionado explican cómo instalar todo ese software

si aun no lo ha hecho. También vamos a suponer que el núcleo está compilado con el soporte TCP/IP. Para incluir el soporte de red, tendrá que contestar afirmativamente a la pregunta correspondiente que se le hará durante el comando `make config`.

Una vez hecho esto, se deben modificar los ficheros de configuración que usa *NET-2*. Esta parte suele ser bastante simple, pero suele haber bastante desacuerdo entre las diferentes distribuciones de *Linux*. Los ficheros pueden estar en `/etc` o en `/usr/etc` o incluso `/usr/etc/inet`. En el peor caso puede usar el comando `find` para localizar los ficheros. A veces los ficheros están también repartidos por varios directorios y no en uno solo. Lo siguiente es fundamentalmente aplicable a conexiones *Ethernet*. Si lo que va a usar es SLIP, léase esta sección para ir entendiendo los conceptos y luego vea las instrucciones específicas para SLIP.

3.4.3 La configuración de red

Antes de configurar su sistema con TCP/IP necesita conocer cierta información sobre la red. En muchos casos, el administrador local se la proporcionará.

- ✓ Dirección IP. Es la dirección única de cada máquina, formada por números separados por puntos. Por ejemplo, 128.253.153.54. El administrador de red le dará este número.

Si está configurando el modo "*loopback*" únicamente (esto es, no tiene conexión a la red mediante SLIP o *Ethernet*) su dirección IP será la 127.0.0.1. o Máscara de red ("*netmask*"). Es un número similar a la dirección IP, que determina qué parte de la dirección IP determina el número de sub-red, y qué parte especifica el host en la sub-red (si todo esto no lo comprende bien, le sugerimos que lea documentos sobre administración de red).

Algunas de las cosas que aquí se exponen proceden del documento *NET-2 HOWTO* de Terry Dawson y Matt Welsh.

La máscara de red es un patrón de bits, que al ser superpuesto a una dirección de la red, le dirá en qué sub-red se encuentra esa dirección. Esto es muy importante para el rutado y, si usted nota que puede comunicar con gente de redes externas pero no con gente de su misma red, es un buen motivo para pensar que tiene mal puesta la máscara.

Los administradores de la sub-red habrán seleccionado las máscaras en tiempo de diseño de la red, y serán quienes deban darle esa información. Muchas sub-redes son de "clase C" y usan la máscara 255.255.255.0. Otras sub-redes de "clase B" usan la 255.255.0.0. El código de NET-2 seleccionará automáticamente una máscara que asume que no hay subred.

Todo esto debe aplicarse también a la configuración "*loopback*". Dado que la dirección "*loop-back*" es siempre la 127.0.0.1, la máscara será la 255.0.0.0. Puede especificarla de forma explícita o dejar que el sistema la ponga por defecto.

✓ Dirección de red. Es el resultado de la operación lógica AND entre su dirección IP y la máscara.

Por ejemplo, si su dirección IP es la 128.253.154.32 y la máscara es 255.255.255.0, su dirección de red será la 128.253.154.0. Con una máscara 255.255.0.0, la dirección sería 128.253.0.0. Si utiliza solo la configuración en "*loopback*", la dirección de red no existe.

✓ Dirección de "broadcast". Se utiliza para lanzar paquetes que deben recibir todas las máquinas de la subred. Así pues, si el número de host de la subred se obtiene mediante el último octeto de la dirección IP (o sea, la máscara es la 255.255.255.0), su dirección de "*broadcast*" será su dirección de red operado en OR con 0.0.0.255.

Por ejemplo, si su número IP es el 128.253.154.32, y la máscara es la 255.255.255.0, la dirección de "*broadcast*" sería la 128.253.154.255.

Observe que por motivos históricos, algunas subredes están configuradas para usar la dirección de red como dirección de "*broadcast*". Si tiene dudas, consulte con el administrador de la red. En muchos casos, bastará con copiar la configuración que tengan

otras máquinas de la subred y cambiar únicamente el valor IP, por supuesto. La dirección "*broadcast*" tampoco tiene utilidad en una configuración en "*loopback*".

- ✓ Dirección de pasarela. Se trata de la dirección de la máquina que va a ser su pasarela a otras máquinas que no estén en su misma subred. Muchas veces es una dirección IP como la suya, solo que terminada en ".1". Por ejemplo, si la dirección IP es la 128.253.154.32, la de la pasarela podría ser la 128.253.154.1. El administrador se la dirá en cualquier caso.

En ocasiones puede tener varias pasarelas. Una pasarela o *gateway* es simplemente una máquina que se encuentra a la vez en dos subredes (tiene una dirección IP por cada una), y reparte los paquetes entre ellas. En muchas subredes existe una sola pasarela para comunicarse con las redes externas, pero en otras hay varias, una para cada subred adicional. Si su red está aislada de otras, o su máquina se encuentra en configuración "*loopback*", no necesitará dirección de pasarela.

- ✓ Dirección del servidor de nombres. Suele existir un servidor que traduce nombres de máquinas a direcciones IP. El administrador le facilitará su dirección. Puede usted mismo ejecutar en su máquina un servidor de nombres, el programa *named*, en cuyo caso su dirección será la 127.0.0.1. A menos que realmente lo necesite, le recomendamos que procure siempre usar otra máquina distinta. La configuración de *named* es otro tema; y lo primordial aquí es que comunique con la red. Puede tratar estos asuntos más tarde.

En una configuración "*loopback*" no es necesario este dato. Nota para usuarios de SLIP: La información anterior puede necesitarla o no.

Cuando use SLIP su dirección IP será determinada de dos formas: bien "estática", lo que significa que será siempre a misma, o bien "dinámica", lo que indica que le será asignada una de las disponibles cada vez que conecte con el servidor SLIP.

NET-2 implementa rutado completo, múltiples rutas, subredes... Lo anterior describe las configuraciones más básicas. Pero la suya puede ser diferente: cuando tenga alguna

duda, consulte al administrador de la red, y eche un vistazo a las páginas del manual para `route` e `ifconfig`. La configuración completa de redes TCP/IP supera ampliamente las intenciones de esta guía, y con lo anterior sólo pretendemos posibilitar que todo el mundo pueda poner en marcha su sistema en una red ya configurada.

- ✓ Los ficheros de inicio `rc` para trabajo en redes: Los ficheros `rc` son *shell scripts* que se ejecutan durante el arranque del sistema para configurarlo. Son ejecutados por el proceso `init`, y ponen en marcha los demonios básicos como `sendmail` o `cron` y además configuran parámetros de la red como la dirección IP y el nombre del `host`. Estos *scripts* se suelen encontrar en `/etc/rc.d` o en `/etc`. Lo que vamos a hacer aquí es describir los ficheros `rc` que configuran TCP/IP.

En Linux son dos: `rc.inet1` y `rc.inet2`. El primero configura parámetros básicos como direcciones IP e información de rutado. El segundo lanza los demonios TCP/IP, principalmente `inetd`, quien se encargará de lanzar cuando haga falta los `telnet` y demás. En muchos sistemas se juntan los dos ficheros en uno, el `rc.inet` o `rc.net`. No tiene importancia el nombre, siempre que se ejecuten en el momento adecuado durante el arranque. Para conseguirlo, `init` tiene que saberlo, y para ello existen entradas específicas en el fichero `inittab`. En el peor caso tendría usted que crear las entradas para `rc.inet1` y `rc.inet2` en dicho fichero.

Como hemos dicho, `rc.inet1` configura los parámetros básicos de red. Esto incluye el número IP y dirección de red, y la tabla de rutado. Estas tablas se usan para rutar los datagramas entrantes y salientes de otras máquinas. Lo más simple es tener tres rutas: una para enviar paquetes a su propia máquina, otra para enviarlos a otra máquina de la subred y una tercera para enviarlos a máquinas de otras subredes (a través de una pasarela). Para configurar esto se usan los programas `ifconfig` y `route`, programas que suelen estar en `/etc`.

`ifconfig` se utiliza para configurar el dispositivo interfaz de red con los parámetros que necesita, como la dirección IP, la máscara, dirección de `broadcast` y otros, `route` por su lado, se utiliza para crear o modificar entradas de la tabla de rutado.

Para muchas configuraciones, el siguiente rc.inet1 puede valer, aunque, por supuesto, necesitará poner sus propias direcciones IP y demás.

```
#!/bin/sh
# /etc/rc.d/rc.inet1 -- Configuración de TCP/IP
# Configuración del dispositivo 'loopback'
HOSTNAME=`hostname`
/etc/ifconfig lo 127.0.0.1 # utiliza por defecto la máscara 255.0.0.0
/etc/route add 127.0.0.1 # una ruta apunta al dispositivo 'loopback'
# Configuración del dispositivo Ethernet. Si solo se usa el 'loopback',
# comentar las líneas siguientes.
# EDITELO con sus propios datos.
IPADDR="128.253.154.32" # PONGA aquí su dirección IP
NETMASK="255.255.255.0" # PONGA aquí su máscara de red
NETWORK="128.253.154.0" # PONGA aquí su dirección de red
BROADCAST="128.253.154.255" # PONGA aquí su dirección 'broadcast' si
# la tiene. Si no, elimine la línea.
GATEWAY="128.253.154.1" # PONGA aquí su dirección de pasarela
/etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}
# Si no tiene dirección de 'broadcast', ponga la anterior línea así:
# /etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK}

/etc/route add ${NETWORK}
# Lo que sigue solo hace falta si hay pasarela, o sea, si su subred esta
# conectada a otra red.
/etc/route add default gw ${GATEWAY} metric 1
# Fin de la configuración de Ethernet
```

Quizás tenga que estudiarse un poco más el tema para su instalación particular, aunque en la mayor parte de los casos el fichero anterior será suficiente.

rc.inet2 arranca servidores usados por TCP/IP. El más importante es inetd, que queda en segundo plano y escucha por varios puertos de la red. Cuando una máquina intenta

conectarse por uno de ellos (por ejemplo, por el de telnet), inetd envía una copia del servidor correspondiente (en este caso, in.telnetd) para que controle el puerto afectado. Esto es mejor que mantener en ejecución todos los servidores de red necesarios (múltiples copias de in.telnetd, in.ftpd y demás).

inetd los arranca conforme se van necesitando. Pero en rc.inet2 se arrancan también otros demonios. syslogd se ocupa de acumular los mensajes generados por el núcleo y diversas aplicaciones y tratarlos según diga el fichero /etc/syslogd.conf (guardarlos en ficheros, sacarlos por consola, ..). routed se ocupa de la información de rutado dinámica. Cuando su sistema intenta enviar paquetes a otra red, puede requerir nuevas entradas en las tablas de rutado, que routed trata sin necesidad de intervención del usuario. El ejemplo siguiente solo arranca un número mínimo de servidores. Existen otros que pueden interesarle, como el NFS. Cuando instale TCP/IP en su sistema, es mejor empezar con una configuración sencilla y luego complicarla según sus necesidades.

Observe que en el fichero siguiente se asume que los servidores de red se encuentran en /etc, pero pueden estar en otro sitio (en /sbin, por ejemplo).

```
#!/bin/sh
# Ejemplo de /etc/rc.d/rc.inet2
# Arrancar syslogd
if [ -f /etc/syslogd ]
then
/etc/syslogd
fi
# Arrancar inetd
if [ -f /etc/inetd ]
then
/etc/inetd
fi
# Arrancar routed
if [ -f /etc/routed ]
then
/etc/routed -q
fi
```

Hecho!

Otro servidor que puede interesarle es *named*, servidor de nombres, que traducirá nombres (locales) a direcciones IP y viceversa. Si no hay servidor de nombres en su subred o quiere proporcionar nombres nuevos a la misma, necesitará arrancar *named*. Su configuración es más compleja y requiere cierto cuidado y planificación, por lo que le recomendamos consultar bibliografía específica.

Sin embargo, no es habitual tener que instalar un servidor de nombres en su sistema.

✓ ***/etc/hosts***

/etc/hosts lleva una lista de direcciones IP y nombres de máquinas que les corresponden. En general, */etc/hosts* solo contiene entradas para su máquina y quizás alguna otra "importante", como servidores de nombres o pasarelas. Su servidor de nombres local proporciona a otras máquinas traducción automática del nombre de su host a su dirección IP.

Por ejemplo, si su máquina es *loomer.vpizza.com* con la dirección IP 128.253.154.32, su */etc/hosts* sería como este:

```
127.0.0.1 localhost
128.253.154.32 loomer.vpizza.com loomer
```

Si solo usa el "*loopback*", la única línea necesaria es la que tiene el número 127.0.0.1, añadiendo tras *localhost* el nombre de su máquina.

✓ ***/etc/networks***

El fichero */etc/networks* tiene direcciones de su red y otras, y es usado por el comando *route*. Permite dar nombre a las redes. Cada subred que quiera añadir a *route* debe aparecer en */etc/networks*. Por ejemplo,

```
default 0.0.0.0 # rutado por defecto - obligatorio
```

```
loopnet 127.0.0.0 # red de 'loopback' - obligatorio
mynet 128.253.154.0 # Ponga aquí su dirección de red
```

✓ **/etc/host.conf**

Este fichero dice a su sistema cómo resolver los nombres de los hosts. Debe contener dos líneas:

```
order hosts,bind
multi on
```

Estas líneas indican a los mecanismos de resolución que empiecen buscando en el fichero /etc/hosts y luego pregunten al servidor de nombres, si existe. La entrada multi permite que para un nombre de máquina haya varias direcciones IP en /etc/hosts.

✓ **/etc/resolv.conf**

En este fichero se configura el mecanismo de resolución, especificando la dirección del servidor de nombres y el nombre del dominio de su máquina. El dominio es como un nombre de host "mutilado".

Por ejemplo, si su máquina se llama *loomer.vpizza.com*, el dominio será

```
vpizza.com.
```

Como fichero /etc/resolv.conf de ejemplo, veremos el caso de la máquina goober.norelco.com cuyo servidor de nombres es el 127.253.154.5:

```
domain norelco.com
nameserver 127.253.154.5
```

Con líneas *nameserver* adicionales podrá especificar la existencia de varios servidores de nombres.

3.4.4 Ajuste del nombre de su *host*

Para activar el nombre de su *host* debe usar el comando *hostname*. Esto suele hacerse en un fichero como */etc/rc.local*. Busque en sus ficheros *rc* y busque una llamada a *hostname* como la siguiente:

```
/bin/hostname loomer.vpizza.com
```

Vea que hay que especificar el nombre completo (dominio incluido).

3.4.5 Problemas con la configuración

Una vez que haya preparado los ficheros anteriores, habrá que reiniciar *Linux* para que reconozca las nuevas configuraciones. Luego tendrá que hacer pruebas, para las que lo más indicado es probar aspectos individuales de la red y no tratar de empezar, por ejemplo, lanzando un proceso *Mosaic* con una conexión *X*.

Con el comando *netstat* puede ver las tablas de rutado. Esta suele ser la principal fuente de problemas. En la página del manual para este comando encontrará la sintaxis adecuada. Para comprobar que funciona su conexión, le sugerimos probar un cliente como *telnet* para ver si puede conectarse a máquinas de su subred y de otras redes. Esto puede ponerle sobre la pista del problema.

Por ejemplo, si puede conectarse a máquinas de otras subredes pero no de la suya propia, puede tratarse de un problema con la máscara de red o las tablas de rutado. Ejecutando *route* como *root* podrá jugar directamente con las entradas de la tabla. Para hacer estas pruebas de conectividad, utilice direcciones IP y no nombres. Así, si tiene problemas para ejecutar

```
$ telnet shoop.vpizza.com
```

La causa puede ser una configuración incorrecta del servidor de nombres. Si funciona usando la dirección IP, se puede casi asegurar que el resto de la configuración está bien hecha. Solo falta que funcione bien el servicio de nombres (probablemente haya que especificar correctamente la dirección del servidor de nombres).

La depuración de configuraciones de red puede ser tarea difícil, y no podemos tratarla aquí. Le sugerimos, si no consigue otra ayuda, que consulte el libro *Linux Network Administrators' Guide*.

3.4.6 .Configuración de SLIP

Con SLIP (*Serial Line Internet Protocol*) usted puede conectarse a una red TCP/IP mediante una línea serie, como puede ser un *modem* o una línea dedicada asíncrona. Por supuesto, para usar SLIP tiene que tener acceso a un servidor SLIP. Muchas empresas y universidades proporcionan acceso por poco dinero.

Podemos destacar dos programas relacionados con SLIP: *dip* y *slattach*. Ambos se usan para iniciar una conexión SLIP y por lo tanto son necesarios. No es suficiente con llamar al servidor SLIP con programas como *kermit* y después usar los comandos *ifconfig* y *route*. Esto se debe a que *dip* y *slattach* realizan una llamada especial *ioctl()* para convertir el control de un dispositivo serie a 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 SLIP, por lo que está indicado para líneas dedicadas que no requieren interacción con el módem o similar. Casi todo el mundo, sin embargo, usa *dip*.

Con *dip* también puede configurar su sistema como servidor SLIP, permitiendo a otras máquinas conectarse a la red a través de su *modem* y su conexión *Ethernet*.

A SLIP se le llama conexión "punto a punto" (*point-to-point*) 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*) que también se ha portado a *Linux* .

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 "loopback" o con *Ethernet*. En las siguientes líneas le mostramos las diferencias. Es importante que vea lo que hemos explicado antes sobre configuración en general, y aplique ahora las modificaciones que le vamos a contar.

3.5 Conexiones SLIP con asignación de IP estática usando *dip*

Si su servidor SLIP le permite tener la dirección IP estática, lo más adecuado es insertar la dirección y el nombre del host en el fichero */etc/hosts*. Además, debe configurar los ficheros *rc.inet2*, *hosts.conf* y *resolv.conf* como se ha dicho antes.

En el fichero *rc.inet1* también tendrá que introducir cambios, ejecutando *ifconfig* y *route* solo para el dispositivo "loopback", puesto que *dip* hará lo propio con el dispositivo SLIP.

Pero si usa *slattach* sí tendrá que incluir comandos *ifconfig/route* en *rc.inet1* para el dispositivo SLIP (en breve veremos cómo).

El programa *dip* debería configurar sus tablas de rutado para la conexión SLIP. Sin embargo, puede no hacerlo bien, y tendrá que corregirlo ejecutando por su cuenta *ifconfig* o *route* cuando se haya conectado. Quizás le convenga entonces escribirse un *shell script* para hacerlo automáticamente. En muchos casos, la pasarela es el propio servidor SLIP.

De todas formas, el comando `dip` puede deducirlo de la información que envía el servidor al conectarse.

Puede que necesite el argumento *pointpoint en ifconfig* si ve que `dip` no lo configura bien. Por ejemplo, si la dirección del servidor SLIP es 128.253.154.2 y la suya es 128.253.154.32, el comando a ejecutar (como *root*) podría ser: `ifconfig sl0 128.253.154.32 pointpoint 128.253.154.2` tras conectar con *dip*. La documentación en línea de este comando le será útil. Observe que los dispositivos SLIP que se usan en *ifconfig* y *route* son `sl0`, `sl1`, etc. (y no como en *Ethernet*, que es `eth0`, `eth1`, etc.) En la sección posterior le explicaremos cómo configurar `dip` para conectarse a un servidor SLIP.

3.5.1 Conexiones SLIP con asignación de IP estática usando `slattach`

Si tiene una línea dedicada o un cable conectado directamente al servidor SLIP, no necesitará usar `dip` para iniciar la conexión. En su lugar puede usar *slattach*.

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

```
IPADDR="128.253.154.32" # Ponga aquí su dirección IP
REMADDR="128.253.154.2" # Ponga aquí la del servidor de SLIP
# Modifique lo siguiente para su dispositivo serie
slattach -p cslip -s 19200 /dev/ttyS0
/etc/ifconfig sl0 $IPADDR pointpoint $REMADDR up
/etc/route add default gw $REMADDR
slattach asigna el primer dispositivo SLIP disponible (sl0, 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*. El segundo es un SLIP que incluye compresión de las cabeceras de los data gramas. Por ello su elección habitual será *cslip* a menos que tenga algún problema con la conexión.

Si hay más de un dispositivo SLIP tendrá que considerar 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 del comando *route*, etc.

3.5.2 Conexiones SLIP con asignación de IP dinámica usando *dip*

Si el servidor SLIP le asigna dinámicamente las direcciones IP, no sabrá, evidentemente, su dirección IP antes de conectarse, con lo que no puede incluir esa información en */etc/hosts* (aunque sí incluirá la información de "*loopback*", 127.0.0.1).

Muchos servidores SLIP envían al terminal la dirección IP y la del propio servidor. Por ejemplo, un servidor SLIP podría decirle esto al conectarse:

Your IP address is 128.253.154.44.

Server address is 128.253.154.2.

dip puede capturar ese texto y configurar así el sistema.

Ahora le indicaremos cómo se configura SLIP para conectarse al servidor SLIP.

3.5.3 Utilización de *dip*

dip puede facilitar el proceso de conexión a un servidor SLIP, pues se ocupará de entrar en el sistema remoto y configurar el dispositivo SLIP según la información recibida del servidor. Este programa es el más indicado a menos que su línea sea dedicada.

Para utilizar *dip* tendrá que escribir un "*script*" que contendrá comandos para comunicar con el servidor SLIP durante la entrada en el sistema remoto. Por ejemplo, incluirá envío automático de usuario y *password* al servidor así como lo necesario para asignar la dirección IP.

Lo que sigue es un ejemplo de script para asignación dinámica de dirección IP. Para asignación estática puede poner al principio del script los valores fijos a \$local y \$remote (direcciones IP local y remota, respectivamente). Vea los manuales de dip para más información.

main:

MTU es 'Maximum Transfer Unit' o tamaño máximo de los paquetes

transmitidos por el dispositivo SLIP. En muchos servidores este

valor debe ser 1500 o 1506. Hable con el administrador de la red

si no esta seguro.

web \$mtu 1500

Hacer que el rutado de SLIP sea el de su sistema por defecto.

default

Elegir puerto serie y velocidad.

port cua03

speed 38400

Reiniciar el modem y la línea del terminal. Si le da problemas,

comente la línea.

reset

Ponga aquí su cadena de inicio del modem.

send ATT&C1&D2\\N3&Q5%M3%C1N1W1L1S48=7\r

wait OK 2

if \$errlvl != 0 goto error

Llamar al servidor SLIP (ponga aquí el teléfono).

dial 2546000

if \$errlvl != 0 goto error

wait CONNECT 60

if \$errlvl != 0 goto error

En este punto estaremos conectados. Entrar en el sistema.

login:

sleep 3

send \r\n\r\n

Esperar el 'prompt' de entrada (login).

```

wait login: 10
if $errlvl != 0 goto error
# Enviar su nombre de usuario.
send USERNAME\n
# Esperar el 'prompt' de password.
wait ord: 5
if $errlvl != 0 goto error

# Enviar su password.
send PASSWORD\n
# Esperar el 'prompt' del servidor que indica que esta preparado.
wait annex: 30
if $errlvl != 0 goto error
# Enviar un comando al servidor para empezar la conexión.
send slip\n
wait Annex 30
# Obtener la dirección IP desde el servidor. El comando 'get...remote'
# lee un texto de la forma xxx.xxx.xxx.xxx y lo asigna a la variable
# dada como segundo argumento (aquí es $remote).
get $remote remote
if $errlvl != 0 goto error
wait Your 30
# Obtener la dirección local IP desde el servidor y asignarla a $local.
get $local remote
if $errlvl != 0 goto error
# Establecer la conexión SLIP.
done:
print CONNECTED to $remote at $rmtip
print GATEWAY address $rmtip
print LOCAL address $local
mode SLIP
goto exit
error:

```

```
print SLIP to $remote failed.
```

```
exit:
```

dip ejecuta automáticamente los programas *ifconfig* y *route* según los valores asignados a *\$local* y *\$remote*. Aquí, esas variables son asignadas con el comando *get. .remote*, que obtiene el texto de la dirección del servidor SLIP y lo asigna a la variable.

Si los comandos *ifconfig* y *route* que *dip* ejecuta no funcionan, siempre puede llamarlos por su cuenta desde un shell script tras ejecutar *dip* o modificar las fuentes del propio *dip*. La opción *-v* de *dip* le dará información para depuración generada durante la conexión y le ayudará a averiguar la(s) causa(s) del problema(s).

Ahora, para probar *dip* y abrir la conexión SLIP, escriba un comando como:

```
/etc/dip/dip -v /etc/dip/mychat 2>&1
```

Las explicaciones de esta sección le deberían haber permitido conectarse a la red, bien sea por *Ethernet* o por SLIP. De nuevo le volvemos a recomendar que consulte un libro sobre configuración de redes TCP/IP, en especial si en la red hay configuraciones especiales de rutado o similar.

3.6 Red con UUCP

UUCP (*UNIX-to-UNIX Communication Protocol*) es un viejo mecanismo usado para transferir información entre sistemas *Unix*. Mediante UUCP, los sistemas Unix se comunican con otros (vía *modem*), transfiriendo mensajes de correo, news, ficheros y demás. Si no tiene acceso TCP/IP o SLIP, puede usar UUCP para comunicarse con el mundo. Casi todo el software de correo puede ser configurado para usar transferencias UUCP. De hecho, si tiene algún servidor Internet cercano, puede recibir correo en su sistema de esa red mediante UUCP.

El libro Linux Network Administrator's Guide le dará información completa para configurar utilizar UUCP en Linux . También encontrará información en el documento UUCP-

DOWTO, que puede obtener por FTP anónimo de sunsite.unc.edu. Otra fuente de información sobre UUCP es el libro *Managing UUCP and USENET*, de Tim O'Reilly y Grace Todino.

3.7 Correo Electrónico

Como casi todos los *UNIX*, *Linux* dispone de paquetes de software para tener correo electrónico. Este puede ser tanto local (entre usuarios de su sistema) como remoto (mediante una red TCP/IP o UUCP). El *software* de *E-Mail* consta normalmente de dos partes:

Un agente de usuario o *mailer* y un programa de transporte. El agente de usuario es el software que el usuario utiliza para crear mensajes, leerlos, etc.

Podemos destacar aquí los programas *elm*, *pine* y *mailx*. El programa de transporte es quien se ocupa de entregar correo tanto remoto como local, conociendo protocolos de comunicaciones y demás. El usuario nunca interactúa directamente con este programa, sino que lo hace a través del agente de usuario. Sin embargo, el administrador del sistema debe conocer cómo funciona el programa de transporte, con el fin de configurarlo según sus necesidades.

En *Linux*, el más conocido de los programas de transporte es *Smail*. Es fácil de configurar y capaz de enviar tanto correo local como remoto vía UUCP o TCP/IP. En otros sistemas Unix se suele usar con más frecuencia el programa *sendmail*, que es bastante más complicado de configurar, por lo que no se suele usar en *Linux*.

En el documento *Linux Mail HOWTO* se expone más información sobre el software disponible para correo y cómo configurarlo. Si pretende tener correo remoto, necesitará entender los conceptos de TCP/IP o UUCP (según la red utilizada). Los documentos de UUCP y TCP/IP.

Casi todo el software de correo para *Linux* puede obtenerse mediante FTP anónimo de sunsite.unc.edu en el directorio `/pub/Linux/system/Mail`.

3.8 News y USENET

Linux proporciona también todo lo necesario para tratar las *news*. Puede elegir configurar un servidor de news local, que permitirá a los usuarios poner "artículos" a los diversos "grupos" del sistema en cierto modo, es una forma de discutir. Sin embargo, si tiene acceso a una red UUCP o TCP/IP, podrá participar realmente en *USENET*, una red de news de ámbito mundial. En el *software* de *news* hay dos partes, el servidor y el cliente. El servidor de *news* es el software que controla los grupos de *news* y se ocupa de enviar los artículos a otras máquinas (si estamos en una red). El cliente, o lector de *news*, es el software que conecta al servidor para permitir que los usuarios lean y escriban artículos. Hay varios tipos de servidores de news para *Linux*. Todos siguen un diseño y esquema de protocolos parecido. Principalmente, tenemos los servidores "C News" e "INN". En cuanto a clientes, destacamos *rn* y *tin*. La selección del cliente es cuestión de gustos, y, por supuesto, es independiente del servidor elegido.

Si solo pretende leer y escribir artículos localmente (no como parte de *USENET*), necesitará un servidor que corra en su sistema, así como el lector para los usuarios. El servidor guardará los artículos en un directorio como `/usr/spool/news`, y el lector se compilará para buscar los artículos en ese directorio.

Sin embargo, si desea tener news en red, tendrá ahora varias opciones más.

Para redes basadas en TCP/IP se usa el protocolo NNTP (*Network News Transmission Protocol*). NNTP permite al cliente leer los artículos a través de la red, desde una máquina remota. NNTP también permite a los servidores enviarse artículos por la red. En esto se basa *USENET*. Casi todas las empresas y universidades conectadas cuentan con uno o más servidores NNTP para controlar todas las *news USENET* en ese lugar. Cualquier otra máquina de esa empresa o universidad tendrá un lector de news que accederá al servidor con NNTP. Por ello, solo el servidor NNTP guarda artículos en disco. Los clientes no lo hacen, y siempre tienen que conectar con el servidor para leerlos.

A continuación mostramos algunas situaciones típicas de configuración.

News locales. No hay conexión a red o no se desea tener news en red. En este caso, hay que ejecutar *C News* o *INN* en su máquina, e instalar el lector para leer las news locales.

Con acceso a red TCP/IP y servidor NNTP. Si existe un servidor NNTP ya configurado, puede leer y escribir artículos desde su máquina Linux instalando un lector basado en NNTP (casi todos los lectores tienen opciones de configuración para leer news en NNTP). Por lo tanto, no necesita preocuparse de instalar el servidor o guardar artículos en su sistema. El lector se ocupará de enviarlos a la red. Por supuesto, necesitará configurar TCP/IP y tener acceso a la red.

Tiene acceso a la red TCP/IP pero no hay un servidor NNTP. En este caso, puede instalar un servidor NNTP en su sistema. Además, puede instalarlo para comunicarse con otros servidores NNTP para intercambiar artículos.

Desea transferir *news* con UUCP. Si tiene acceso a UUCP, puede participar en *USENET* de la misma forma. Necesitará instalar un servidor de *news* y un lector.

Además necesitará configurar su software UUCP para transferir los artículos periódicamente a otra máquina con UUCP. En UUCP no se usa el protocolo NNTP, sino que posee su propio mecanismo para transferir artículos.

El único inconveniente de muchos clientes y servidores de news es que deben ser compilados a mano, es decir, no usan ficheros de configuración, sino que se configuran en el momento de compilarlos.

Muchos programas de news "estándares" (disponibles por FTP anónimo en ftp.uu.net, directorio /news) podrían no compilarse en Linux . Los parches que hagan falta se encuentran en sunsite.unc.edu, directorio /pub/Linux /system/Mail (aquí se encuentra también todo el software de correo para Linux). Pueden encontrarse también versiones ya compiladas.

Para más información, léase el documento *Linux News HOWTO* que encontrará en *sunsite.unc.edu*. También encontrará ayuda en el manual *Linux Network Administrator's Guide* de la misma serie de este libro. También puede consultar el libro *Managing UUCP and Usenet*, de *Tim O'Reilly* y *Grace Todino*. Una última posibilidad es el documento "*How*

to become a USENET site", disponible en ftp.uu.net, directorio /usenet/news.announce.newusers.

BIBLIOGRAFIA

NEIL Jenkins y Stan Schat, Redes de Área Local, España, 1996

OLAF Kirch, Guía del Administrador de Redes, Conectiva Linux , Alemania, 1994

Hewlett Packard, Introducción a la impresión en Red, U.S.A, 1993

Instituto Tecnológico de Electrónica y Comunicaciones- Telecom, Redes Locales, Bogotá, 1996.

Internet, varios sitios, tema protocolos.