UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
C U A U T I T L A N

"IMPLEMENTACION DE UNA RED BASADA EN LOS PROTOLOCOLOS TCP/IP Y NFS"

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

MARCO ANTONIO CRUZ MENDOZA

DIRECTOR: FIS. J. JESUS CRUZ GUZMAN

CUAUTITLAN IZCALLI, EDO. DE MEXICO. 1997

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes. nos permitimos comunicar a usted que revisamos la TESIS:
"Implementación de una red basada en los protocolos TCP/IP y NFS".

que presenta el pasante: Marco Antonio Cruz Mendoza
con número de cuenta: 8506411-7 para obtener el TITULO de:
Ingeniero Mecánico Electricista .

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlan Izcalli, Edo. de Méx., a 16 de Junio de 1997.

PRESIDENTE	Fis. José de Jesús Cruz Guzmán.	11-VI-97
VOCAL	Ing. José Luis Rivera López	12-VI-97
SECRETARIO	Ing. Jorge Buendía Gómez	11/VI/97
PRIMER SUPLENTE	Ing. Martha Lilia Urrutia Vargas	4/VI/97
SEGUNDO SUPLENTE	L.A. Valentín Roldán Vázquez	4/VI/97

DEP/VAP/01

A mis Padres :

Dedico este trabajo como un tributo a su esfuerzo, gracias por sus atinados consejos y por su orientación a lo largo de mi vida.

A mis Hermanos :

Por ese compañerismo y unión, que hacen sentir con su compañia y que sus anhelos se vean culminados.

A mi Novia :

Por su apoyo y comprensión, por ser parte importante de mi vida y con tu compañía siempre caminaré por la senda del éxito.

A mi Director de Tesis :

Gracias por el apoyo brindado, las sugerencias y orientaciones para realizar esta tesis, así como por permitirme formar parte de la U.N.A.M. que tanto le debo y que ha sido el aposento de mi desarrollo.

A mis Compañeros, Profesores y Amigos :

Gracias por el apoyo brindado, el entusiasmo otorgado y la amistad desinteresada, que me motivaron para poder llegar a esta meta.

INDICE

INTRODUCCION

El término red significa dos o más computadoras conectadas entre sí. Hay una gran número de razones para unir las computadoras en redes, pero las dos más importantes son: permitir comunicarse a las personas, y compartir recursos.

Internet es una red de redes de computadoras que intercambian información entre sí. De hecho la palabra Internet se deriva del término internetwork, que significa "trabajo entre redes". Una forma fácil de visualizar Internet es imaginarla como una gran nube con computadoras conectadas. Esta nube está cambiando y creciendo de manera constante, conforme se integran nuevas computadoras y redes, y las redes existentes también cambian.

Una vez en Internet, puede enviar mensajes a otras personas en Internet. Incluso puede enviar mensajes a otras personas que usan otras redes conectadas a la Internet.

En cuanto a compartir recursos, muchas veces esta se lleva a cabo por dos programas distintos ejecutandose en computadoras diferentes. Uno de los programas, llamado servidor,proporciona un recurso en particular. El otro programa, llamado cliente, utiliza este recurso. En la Internet, normalmente el término "cliente" y "servidor" hace referencia a los programas que solicitan y proporcionan los servicios.

Internet está construida sobre una colección de redes que recorren el mundo. Estas redes conectan diferentes tipos de computadoras, y de alguna manera, algo debe mantenerlas a todas unidas. Ese algo es TCP/IP.

TCP/IP es una gran familia de protocolos que se utilizan para organizar las computadoras y dispositivos de comunicaciones en una red. Los dos protocolos más importantes son TCP e IP. IP (Internet Protocol) transmite los datos de un lugar a otro, mientras que TCP (Transmission Control Protocol) asegura que todo funcione correctamente.

Esta tesis aborda el problema de configurar y manejar el software de red TCP/IP en sistemas de computadora UNIX. TCP/IP es uno de los paquetes de software que dominan actualmente las comunicaciones de datos en UNIX. Juega un papel particularmente importante como el software de comunicaciones para redes de área local en UNIX.

Primero discutiremos lo básico de TCP/IP y como se mueven los datos a través de la red. Terminando con la explicación de como configurar y correr TCP/IP en un sistema UNIX. Como caso práctico se presenta la configuración de los principales servicios de red con que cuenta hoy la Facultad de Estudios Superiores Cuautitlán.

CAPITULO 1 Esbozo de TCP/IP

Un modelo de arquitectura desarrollado por la International Standards Organization (ISO) es acostumbrada frecuentemente a describir la estructura y función de protocolos de comunicaciones de datos. Este modelo de arquitectura, llamado Open Systems Interconnect (OSI),provee una referencia común para discutir comunicaciones. Los términos definidos por este modelo están bien especificados y ampliamente utilizados en la comunicación de datos, de hecho, es difícil el discutir la comunicación de datos sin utilizar terminología de OSI.

El Modelo de Referencia de OSI contiene siete capas que definen las funciones de protocolos de comunicación de datos. Cada capa del modelo OSI desempeña una función cuando los datos son transferidos entre aplicaciones cooperativas a través de una red. La figura 1.1 identifica cada capa por nombre y provee una descripción funcional corta para ello.

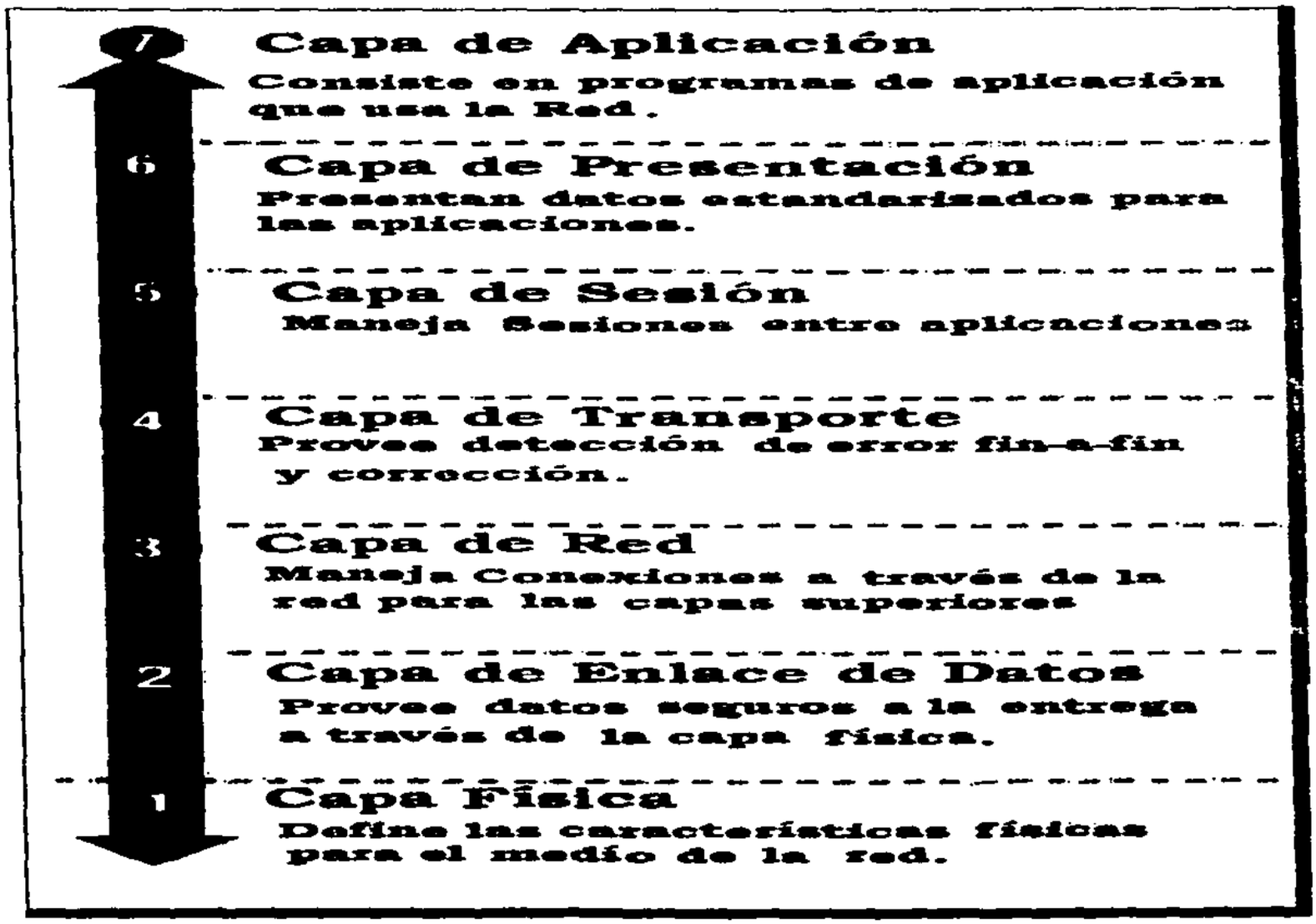

Figura 1.1. El Modelo de Referencia OSI.

Una capa no define un único protocolo define una función de comunicaciones de datos que puede ser desempeñada por cualquier número de protocolos. Por lo tanto, cada capa puede contener múltiples protocolos, cada uno ofrecer un servicio apropiado a la función de esa capa.

Cada protocolo se comunica con su par. Un par es una implementación del mismo protocolo en la capa equivalente en un sistema remoto; por ejemplo, el protocolo de transferencia de archivo local es el igual de un protocolo de transferencia de archivo remoto. Las comunicaciones de una capa igual tienen que estar normalizado para que exitosas comunicaciones tengan lugar. En resumen, cada protocolo es solamente concernido para comunicarse con su igual; no se interesa por la capa anterior o debajo de el.

Sin embargo, tiene que también estar de acuerdo en como pasa información entre las capas en una computadora única, debido a que cada capa está involucrada en enviar datos de una aplicación local a una aplicación remota equivalente. Las capas superiores confían en las capas inferiores para transferir la información sobre la red subyacente. La información es pasada de una capa inferior a la siguiente, hasta que está es transmitida en la red por los protocolos de la Capa Física. En el fin remoto, la información es pasada sobre una capa inferior a la siguiente hasta la aplicación a recibir. Las capas individuales no necesitan conocer como las capas anteriores y debajo a ellas funcionan: únicamente necesitan conocer como pasan información a ellas. El aislar el funcionamiento de las comunicaciones de red en diferentes capas minimizan el impacto de cambio tecnológico en la pila de protocolo completa. Nuevas aplicaciones pueden ser añadidas sin cambiar la red física, y nuevo hardware de red puede estar instalado sin corregir el software de aplicación.

1.1. Arquitectura del Protocolo TCP/IP

Mientras no hay acuerdo universal acerca de como describir TCP/IP como un modelo en capas, está generalmente visto como compuesta de menos capas que los siete utilizado en el modelo OSI. La mayoría de las descripciones de TCP/IP define tres a cinco capas funcionales en la arquitectura de protocolo. Las cuatro capas del modelo son ilustrados en la Figura 1.2 basado en las tres capas (Aplicación, Transporte, y Acceso a la Red), con la adición de la Capa Internet separada. Este modelo provee una representación pictórica razonable de las capas en la pila de protocolo de TCP/IP.

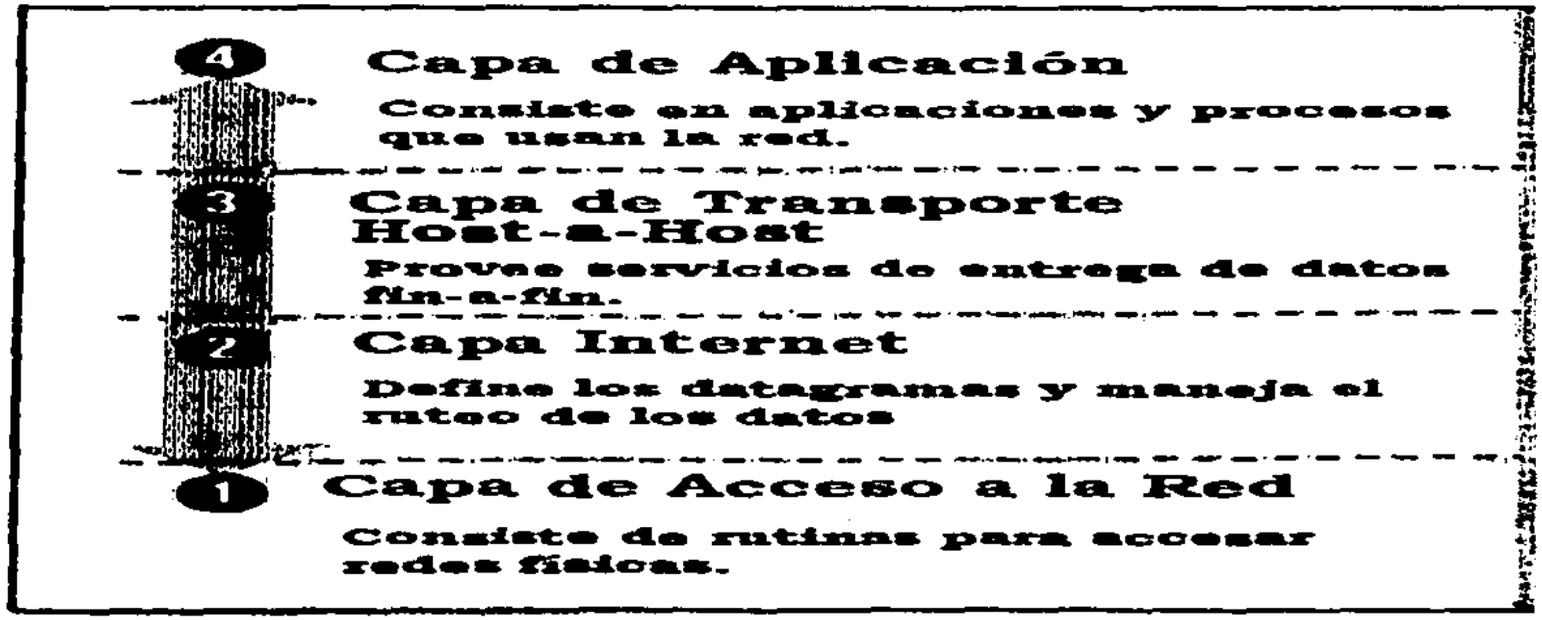

Figura 1.2. Capas en la Arquitectura del Protocolo TCP / IP.

Como en el modelo OSI, los datos son pasado abajo cuando estos se envía a la red, y hacia arriba cuando está los recibe de la red. TCP/IP esta estructurado en cuatro capas que es visto en el modo que es manejado los datos cuando pasa el protocolo abajo de la Capa de Aplicación a la red física subyacente. Cada capa le añade información de control para garantizar la entrega adecuada. Esta información de control es llamado un encabezado porque está situado frente a los datos que son transmitidos. Cada capa trata toda la información que recibe de la capa anteriormente como datos y su propio encabezado frente a esa información. La adición de información de entrega a cada capa es llamado encapsulamiento (la figura 1.3 ilustra esta). Cuando los datos son recibidos, ocurre lo contrario. Cada capa tira fuera su encabezado antes de pasar la información a la capa siguiente. Así como el flujo de la información se amontona, la información recibida de una capa más baja es interpretada tanto como un encabezado como datos.

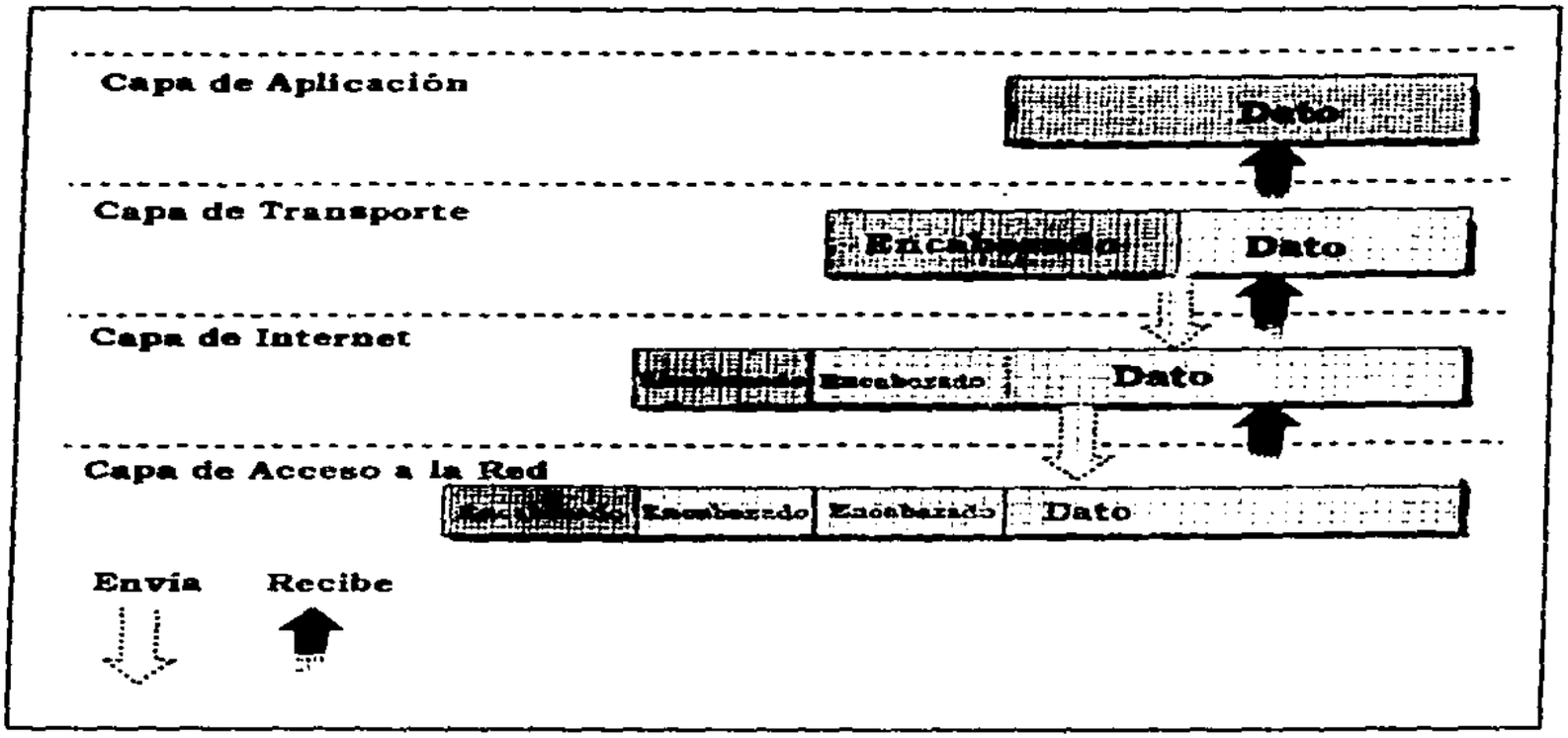

Figura 1.3. Encapsulamiento de Datos.

Cada capa tiene sus propias estructuras de datos independientes. Conceptualmente una capa está inconsciente de las estructuras de datos utilizadas por las capas anteriormente y debajo de ella. En realidad, las estructuras de datos de una capa están diseñado para ser compatible con las estructuras utilizadas por las capas que las rodean para beneficio de transmisión de datos más eficiente. Aún, cada capa tiene sus propias estructura de datos y sus propia terminología para describir esas estructura.

La figura 1.4 muestra los términos utilizados por diferentes capas de TCP/IP para referirce a los datos que se transmiten. Las aplicaciones utilizando TCP refieren un dato como un stream, mientras aplicaciones utilizando el User Datagram Protocol (UDP) refieren un dato como un mensaje. TCP llama un dato segmento, y UDP llama su estructura de datos un paquete La capa de Internet visualiza toda información como bloques llamados datagramas. TCP/IP utiliza muchos diferentes tipos de redes de datos subyacentes para ser transmitidos cada uno de ellos tiene una diferente terminología para la información que transmite. La mayoría de las redes refieren a los datos transmitidos como tramas o frames. En nuestra figurar suponemos una red que transmite piezas de datos llamados frames. Cada uno de estos términos se refieren a

la misma cosa. Los términos varían como es visto de la misma forma como varía la información de capa a capa.

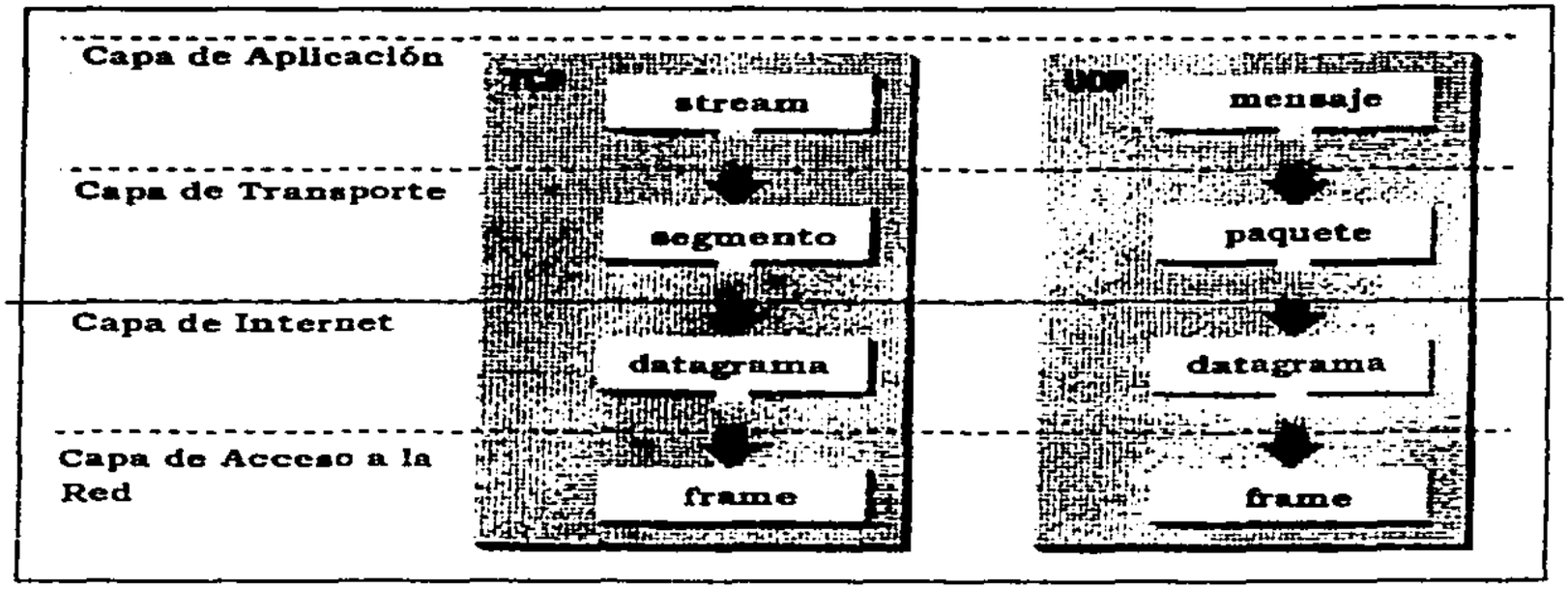

Figura 1.4. Estructura de Datos.

1.2. Capa de Acceso a la Red

La Capa de Acceso a la Red es la más baja capa de la pila de protocolo de TCP/IP. Los protocolos en esta capa proveen los medios para que el sistema suministre información a los otros dispositivos en una red directamente incluidos. Define como utilizar la red para transmitir un datagrama IP. A diferencia de los más altos niveles de protocolos, los protocolos de Capa de Acceso a la Red tienen que conocer los detalles de la red subyacente (su estructura de paquete, direccionamiento, etc.) el correcto formato de la información que se transmite para acatar las restricciones de red. La Capa de Acceso a la Red de TCP/IP sabe acerca de las funciones de las tres capas más bajas del modelo de referencia de OSI (Red, Enlace de Datos, y Físico).

Como nuevas tecnologías de hardware aparecen, nuevos protocolos de Acceso a la Red tienen que ser desarrollados de modo que redes de TCP/IP pueden utilizar el nuevo hardware. Por consiguiente, hay muchos protocolos de acceso por cada estándar de red física.

Funciones desempeñadas a este nivel incluye encapsulamiento de datagrams IP en los frames transmitidos por la red, y el convertir direcciones IP a las direcciones físicas utilizadas por la red. Una de las fortalezas de TCP/IP es este plan de direccionamiento suyo que identifica cada host en el Internet. Esta dirección IP tiene que estar convertida en cualquier dirección apropiada por la red física sobre que el datagrama es transmitido.

Como es implementado en UNIX, los protocolos en esta capa frecuentemente aparecen como una combinación de controladores de dispositivos y programas relacionados. Los módulos que están identificados como nombres de dispositivo de red generalmente encapsulan y suministra la información a la red, mientras programas separados desempeñan funciones relacionadas tal como convertir direcciones.

1.2.1. Ethernet e IEEE 802.3

Ethernet usa el protocolo Carrier Sense Multiple Access with Collision Detect (CSMA/CD). En términos simples: antes de que una estación envíe un paquete, se verifica para ver si otra estación esta activa (detección de portadora). Si es así, espera hasta que el cable este libre, de otra manera es transmitido inmediatamente. Cuando dos o mas estaciones deben empezar el cambio de transmisión a la misma vez (colisión), ellos reconocen esta situación, puesto que ellos siempre comparan el dato enviado con el dato en el cable. Si el dato es corrompido, el procedimiento de transmisión es inmediatamente interrumpido y repetido después de un breve tiempo de espera. El tiempo de espera es determinado por un numero aleatorio generado el cual, en la retransmisión, regresa exponencialmente incrementando el valor de retardo. Esto provoca que dos estaciones usen el mismo tiempo de espera para siempre.

El direccionamiento usa direcciones de 48 bits, las cuales son fijadas en el hardware de cada controlador Ethernet. Puesto que IEEE distribuye números a manufacturadores de controladores Ethernet, es cierto que cada estación en el mundo tiene una única dirección.

La estructura de un paquete Ethernet es mostrada en la figura 1.5.

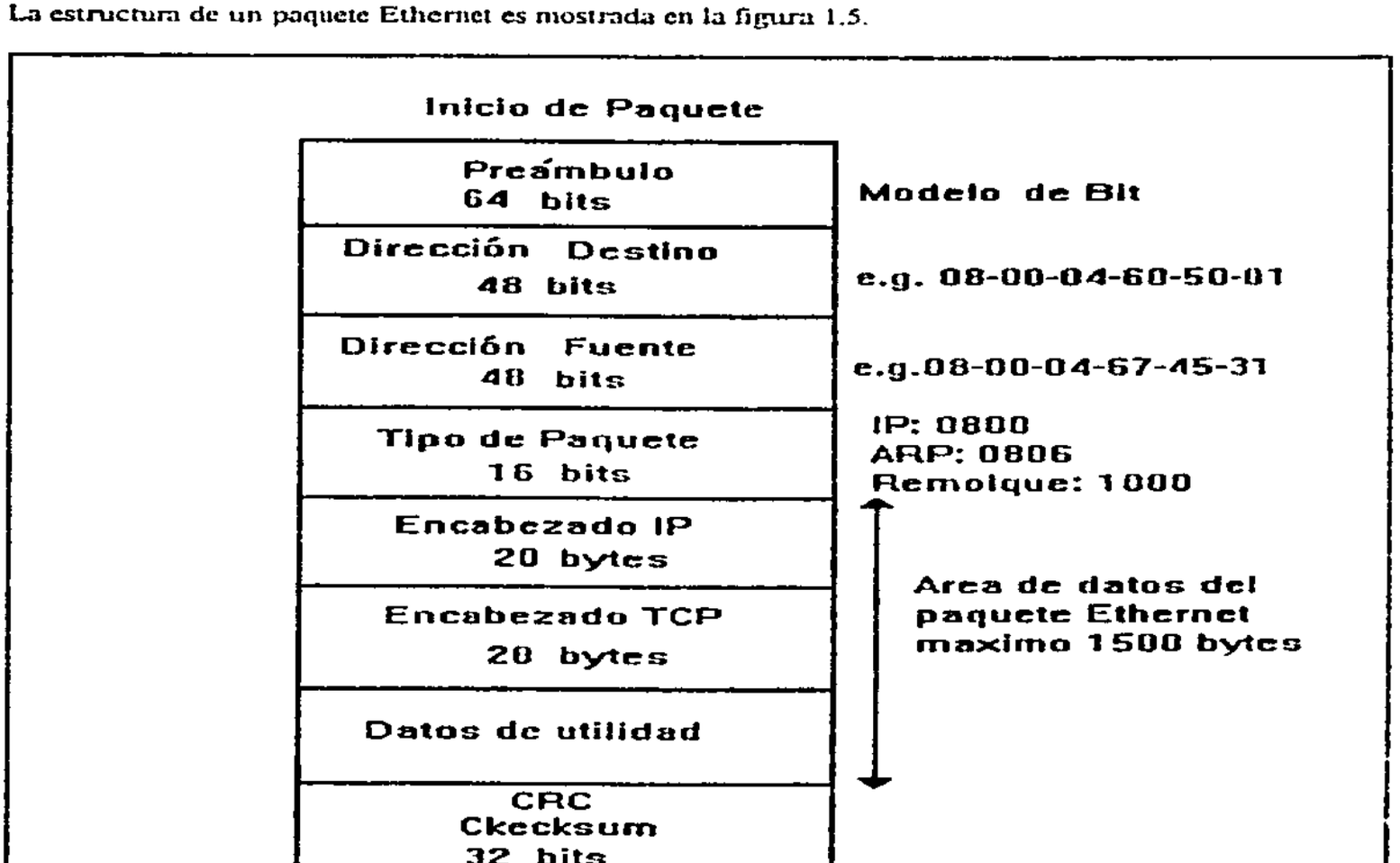

Figura 1.5. Estructura de un Paquete Ethernet para TCP/IP.

El preámbulo para un paquete Ethernet es un modelo de bit especial el cual sirve para sincronizar la estación de recepción. Este es seguido por las direcciones del destinatario y fuente. En el estándar Ethernet, las direcciones de los siguientes protocolos es introducida (en nuestro caso, este es usualmente IP) en el campo del tipo de paquete. Así, varias capas de protocolos de arriba en Ethernet puede ser simultáneamente soportados. En el estándar IEEE 802.3, el campo de tipo de paquete es usado como un campo de longitud; el direccionamiento de las capas altas es llevado afuera en la sección de datos. Después de la sección de datos de máximo de 1500 bytes viene el CRC checksum de 32 bits. Note que por razones técnicas asociadas con la transmisión, un paquete puede ser por lo menos de un largo de 64 bytes. Si es menor el dato a ser enviado, el paquete es artificialmente hecho arriba de este largo por el controlador de Ethernet en el sistema operativo.

Cada estación lee los paquetes en el bus del cable y compara la dirección del destino con el propio. Si los dos son el mismo, las estaciones destino reciben el paquete completamente. Porque este es el propietario (todas las estaciones en la red leen cada paquete), un mensaje puede ser simultáneamente enviado a todas las estaciones (broadcast) o a un grupo de estaciones (multicast). Después veremos como broadcast es usada en un protocolo que será descrito.

Ethernet es muchas veces criticado por la posibilidad de colisiones, lo cual sin embargo es poco frecuente en la practica debido a las altas velocidades de transmisión y el procedimiento de detección de portadora.

1.2.1.1. Direccionamiento de la Capa de Red

Dentro del orden de las direcciones de la capa de red, el encabezado de protocolo Ethernet contiene más de una entrada de dirección. Para IP, RFC 894 especifica una entrada del valor 800(base16) en el campo tipo de paquete Ethernet. RFC 894, es un estandar para la transmición de datagramas IP atraves de redes Ethernet, el cual especifica como los datagramas IP son encapsulados para transmitirse atraves de redes Ethernet.

El estándar 802 de Ethernet prescribe el uso del protocolo Logical Link Control (LLC) en redes de la familia 802. El encabezado del protocolo LLC contiene las direcciones del siguiente protocolo alto. En realidad, LLC no ha penetrado en el mundo de UNIX y todas las implementaciones actuales usan el modo especificado en RFC 894.

1.2.1.2. ARP y RARP

Como se esbozo en secciones anteriores, Ethernet, como muchas otras redes, no usa direcciones Internet usa su propio sistema. Todavía peor, direcciones de Ethernet usan 48 bits que son mas largos que direcciones de Internet (32 bits) y su directa asociación es imposible. El Address Resolution Protocol (ARP) fue diseñado para convertir dinámicamente direcciones IP a direcciones Ethernet, implicando interrogación en la red. En esta forma, por ejemplo, alteraciones en las direccciones Ethernet se hacen transparentes en la red. esto solo puede ser significativamente implementada en una red de área local con broadcast. ARP fue especificado en RFC 826 y ahora es soportado en todos los productos basados en Ethernet.

Los campos hardware, protocolo y operación en la figura 1.6 especifica el tipo de dirección de hardware, la dirección de software (aquí IP) y el tipo de mensaje (pregunta o respuesta).

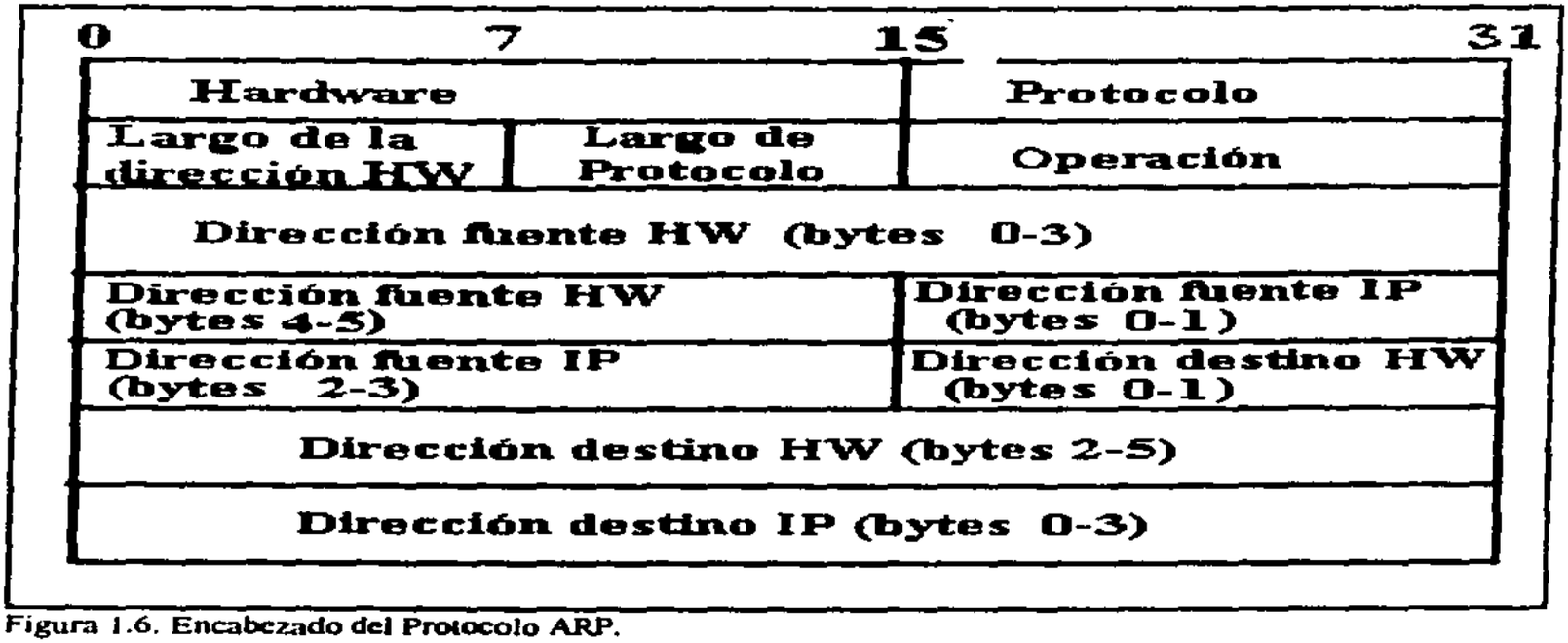

Figura 1.6. Encabezado del Protocolo ARP.

La descripción funcional de ARP es como sigue:

- IP presenta un datagrama en la interfase de red de Ethernet de una computadora A, la cual busca direcciones correspondientes Ethernet en su propia tabla (temporalmente). Si hay entradas validas, un paquete Ethernet con esta dirección es provisto y enviado.
- Si la entrada no es valida, un paquete broadcast ARP con la dirección Internet del host destino es generado y enviado.
- Todas las computadoras en la red reciben un paquete ARP y comparan la dirección Internet contenida con la suya. La computadora (B) con la misma dirección envía una respuesta ARP conteniendo la dirección Ethernet deseada antes en la computadora A.
- La computadora A mete la dirección de la computadora B en esta tabla y envía el paquete IP y los subsecuentes paquetes IP directo a la computadora B.

Para hacer la asociación mas dinámica, las entradas en la tabla temporal son borradas despues de cierto tiempo (1 a 20 minutos, dependiendo de la implementación) y la interrogación es repetida. Puesto que un procedimiento de búsqueda solo lleva unos pocos milisegundos, este procedimiento es prácticamente transparente.

1.3. Capa de Internet

La capa sobre la Capa de Acceso a la Red en la pila de protocolo es la Capa de Internet. El Internet Protocol, RFC 791, es el corazón de TCP/IP y el protocolo más importante en la Capa de Internet. IP provee el servicio de entrega de paquete básico en el cual redes de TCP/IP están construidas. Todos los protocolos, en las capas encima y debajo de IP, utiliza el Internet Protocolo para suministrar datos. Todo el flujo de datos de TCP/IP pasan atraves de IP, entrando y saliendo, independientemente de su destino final.

1.3.1. Internet Protocol

Pero antes de describir estas funciones en más detalle, vamos a mirar algunas de características de IP. Primero, IP es un protocolo sin conexión. Esto significa que IP no hace intercambios de control de información (llamada un "handshake") para establecer una conexión punto-a-punto antes de transmitir datos.

IP confía en protocolos en las otras capas para proveer detección y recuperación de errores. El Internet Protocol es llamado a veces un inseguro protocolo porque no contiene detección de error y recuperación de código. Esto es de no decir que el protocolo de IP no pueda ser confiable por el contrario. IP puede ser confiable en suministrar exactamente sus datos a la red conectada, pero no revisa si la información estuvo recibida correctamente. Los protocolos en otras capas de la arquitectura de TCP/IP provee esta revisión cuando está es requerida.

1.3.1.1. Datagrama

El datagrama es un paquete de formato definido por el Internet Protocol. La figura 1.7 es una representación pictórica de un datagrama IP. Las primeras cinco o seis palabras de 32 bits del datagrama es información de control llamada encabezado. Por omisión, el encabezado tiene cinco palabras de largo; la sexta palabra es opcional. Debido a que la longitud del encabezado es variable, se incluye un campo llamado Internet Header Length (IHL) que indica la longitud de encabezado en palabras. El encabezado contiene toda la información necesaria para entregar el paquete.

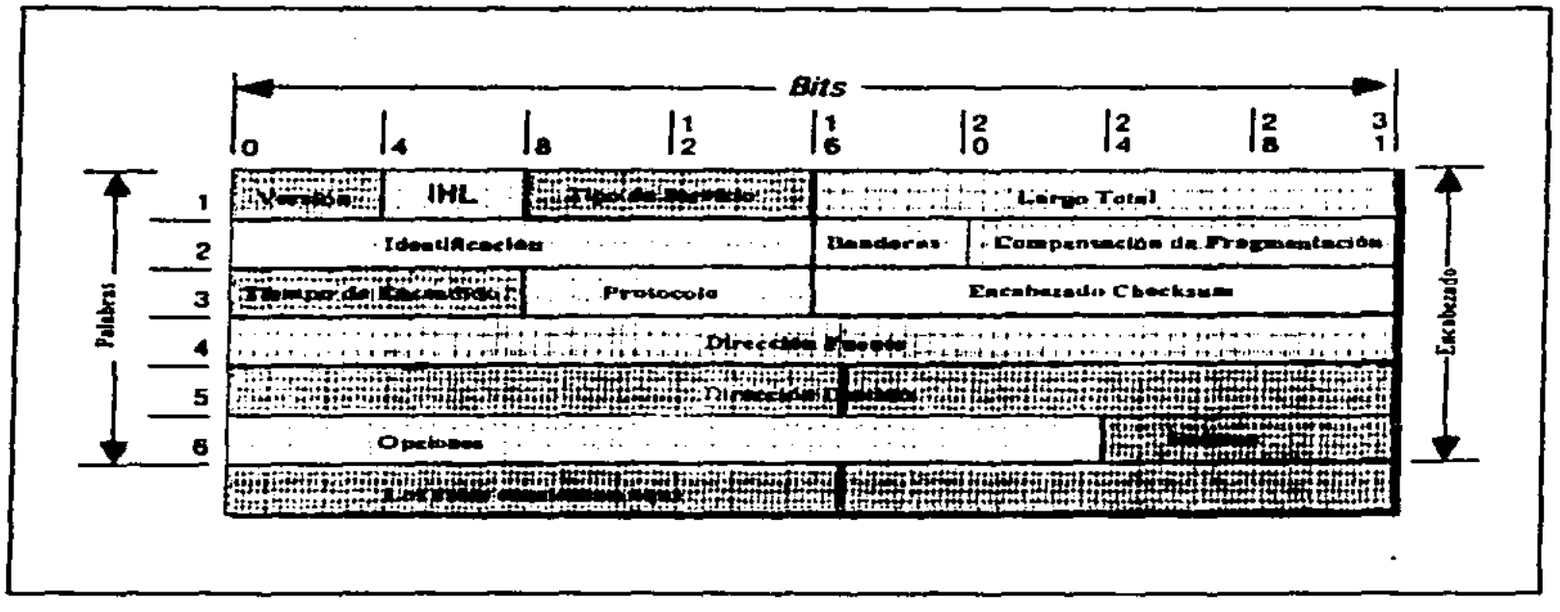

Figura 1.7. Formato de un Datagrama IP.

El Protocolo de Internet suministra al datagrama para checar la dirección de destino la palabra 5 del encabezado. La dirección de destino es una dirección IP estándar de 32 bits que identifica la red de destino y el host especifico en esa red. (El formato de una dirección IP está explicado en el Capítulo 2). Si la dirección de destino es la dirección de un host en la red local, el paquete es entregado directamente al destino. Si la dirección de destino no está en la red local, el paquete es pasado a un gateway para su entrega. Los gateways son dispositivos que intercambian paquetes entre las diferentes redes físicas. El decidir que gateway se utilizara se llama rutear. IP hace la decisión de rutear por cada paquete individual.

1.3.1.2. Ruteando Datagramas

Los gateways de internet son comúnmente (y quizás más exactamente) referido como un ruteador IP porque utilizan el Internet Protocol para rutear paquetes entre redes. En lenguaje técnico común de TCP/IP tradicional, hay solamente dos tipos de dispositivos de red gateways y hosts. Los gateways remiten paquetes entre redes y los hosts no lo hacen. Sin embargo, si un host está conectado en más de una red (llamada un host multi-homed), puede remitir paquetes entre las redes. Cuando un host multi-homed envía paquetes, actúa exactamente como cualquier otro gateway y está considerado como un gateway. La terminología de comunicaciones de datos actual hace a veces una distinción entre gateways y routers, pero vamos a utilizar los términos gateway y ruteador IP recíprocamente.

La figura 1.8 muestra el uso de gateways para remitir paquetes. Los hosts (o sistemas-finales) procesan paquetes a través de todas las cuatro capas de protocolos, mientras los gateways (o sistemas-intermediarios) procesan los paquetes solamente hasta la Capa de Internet donde las decisiones de ruteo son hechas.

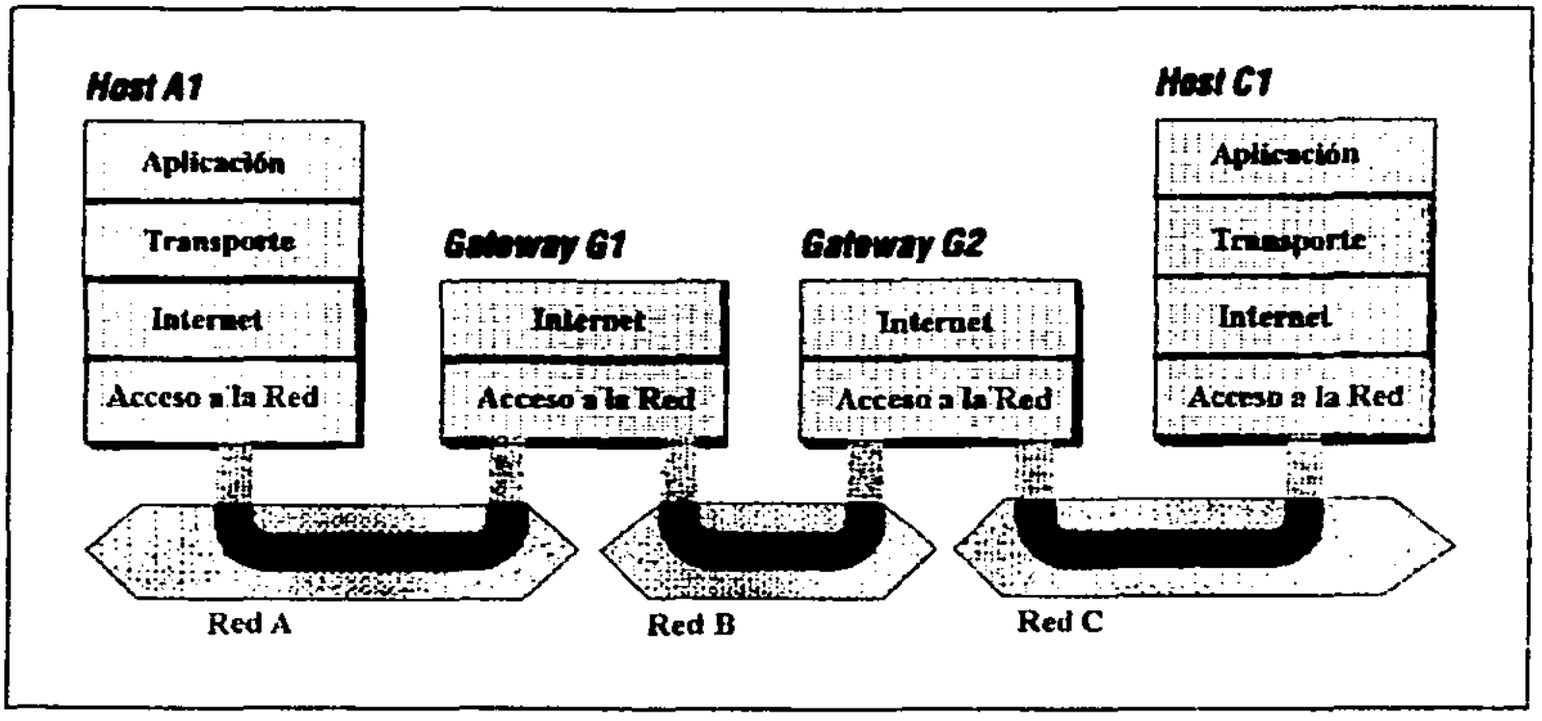

Figura 1.8. Ruteando por Gateways.

La figura 1.9 muestra otra vista del ruteo. Esta figura enfatiza que un datagrama puede viajar a través de diferentes y hasta incompatibles redes físicas subyacentes.

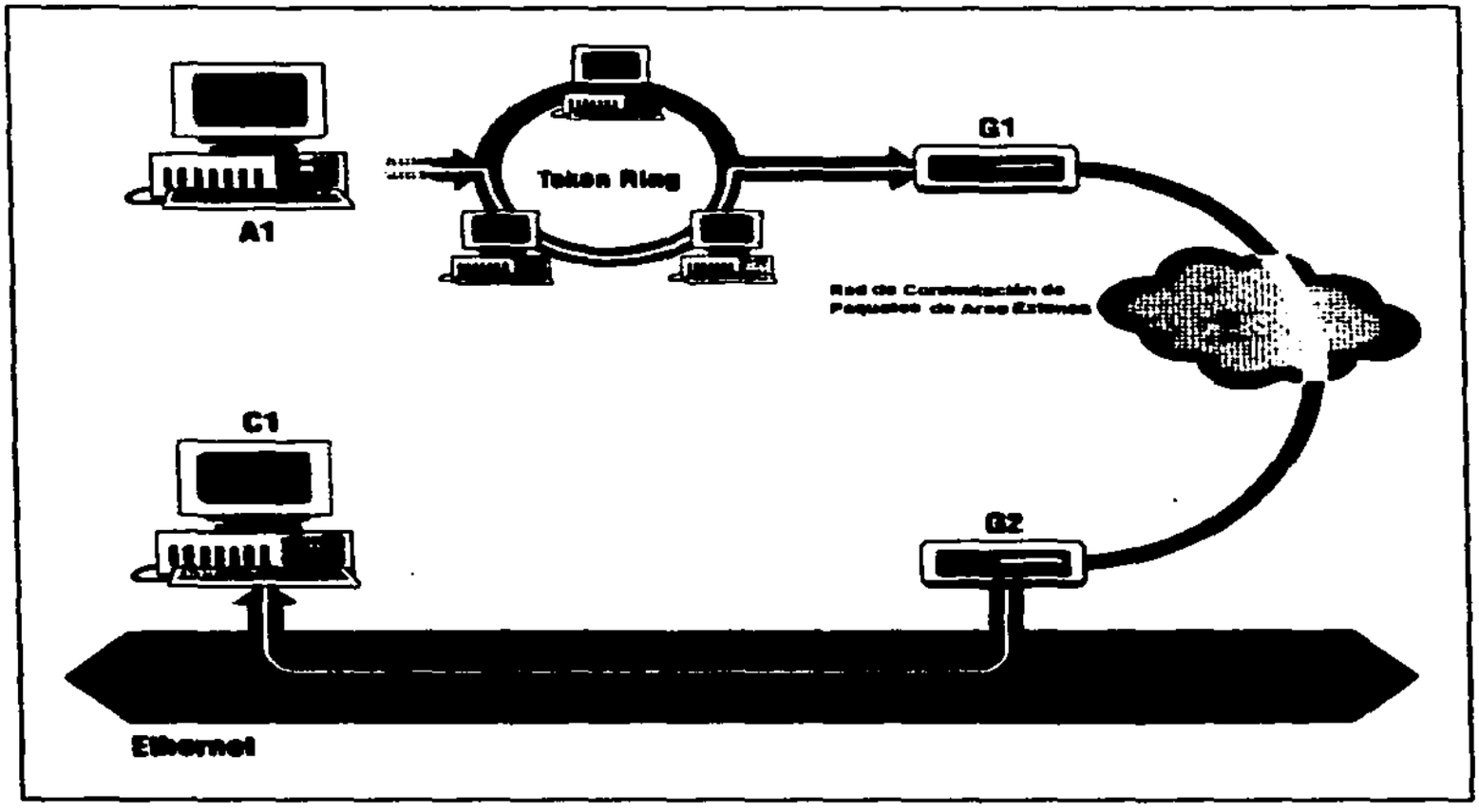

Figura 1.9. Redes, Gateways, y Hosts.

1.3.1.3. Fragmentacion de Datagramas

Como un datagrama es ruteado a través de diferentes redes, puede ser necesario por un módulo de IP en un gateway dividir el datagrama en piezas más pequeñas. Un datagrama recibido de una red puede ser demasiado grande para ser transmitido en un paquete único en una red diferente. Esta condición solamente ocurre cuando un gateway interconecta redes físicas diferentes.

Cada tipo de red tiene un maximum transmission unit (MTU), que es el mayor paquete que puede transferir. Si el datagrama recibido de una red es más largo que la otra MTU de la red, es necesario dividir el datagrama en fragmentos más pequeños para transmisión. Este proceso es llamado fragmentación.

El formato de cada fragmento es el mismo que el formato de cualquier normal datagrama. La segunda palabra del encabezado contiene información que identifica cada datagrama fragmentado y provee información acerca de como reensamblar los fragmentos de vuelta a el datagrama original. El campo Identificación identifica qué fragmento pertenece al datagrama, y el campo Compensación de Fragmentación dice qué pieza del datagrama es este fragmento. El campo Banderas tiene un bit Más Fragmentos que dicen a IP si esta ensamblado todo el datagrama fragmentado.

1.3.1.4. El Paso de Datagramas a la Capa de Transporte

Cuando IP recibe un datagrama que está dirigido al host local, tiene que pasar la porción de datos del datagrama al protocolo de Capa de Transporte correcto. Esto está hecho utilizando el número de protocolo de la palabra 3 del encabezado de datagrama. Cada protocolo de la Capa de Transporte tiene un número de protocolo único que lo identifica IP.

1.3.2. Direccion IP

El Internet Protocol mueve datos entre los hosts en forma de datagramas. Cada datagrama es entregado a la dirección contenida en la Dirección Destino (palabra 5) del encabezado del datagrama. La Dirección Destino es una dirección IP estándar de 32 bits que contiene suficiente información para identificar una red única y un host específico en esa red.

Una dirección IP contiene una parte de red y una parte de host, pero el formato de estas partes no es el mismo en cada dirección IP. El número de bits de dirección utilizados para identificar la red, y el número utilizado para identificar el host, varían de acuerdo a la clase de la dirección. Las tres clases principales de direcciones son: clase A, clase B, y clase C. Examinando los primeros bits de una dirección, el software IP permite determinar rápidamente la clase de la dirección y, por lo tanto, su estructura.

Afortunadamente, esto no es tan complicado como parece. Las direcciones IP se escriben usualmente como cuatro números decimales separados por un punto. Cada uno de los cuatro números se encuentra en el rango de 0-255 (los valores decimales posibles para un solo byte). Debido a que los bits que identifican la clase se encuentran a un lado de los bits de la dirección, podemos juntarlos y ver la dirección como si estuviera compuesta de los bits completos de la dirección de la red y los bits completos de la dirección del host. Un valor inicial del byte:

- Menor que 128 indica una dirección de clase A; el primer byte muestra el numero de la red y los siguientes 3 bytes muestran la dirección del host.
- Entre 128 y 191 indica una dirección de clase B; los primeros 2 bytes identifican la red y los últimos 2 bytes identifican al host.
- Entre 192 y 223 indica una dirección de clase C; los primeros 3 bytes muestran la dirección de la red y el último byte es el número del host.
- Arriba de 223, indica que la dirección es reservada. Podemos ignorar las direcciones reservadas.

La figura 1.10 nos ilustra cómo la estructura de las direcciones varía de acuerdo a la clase.

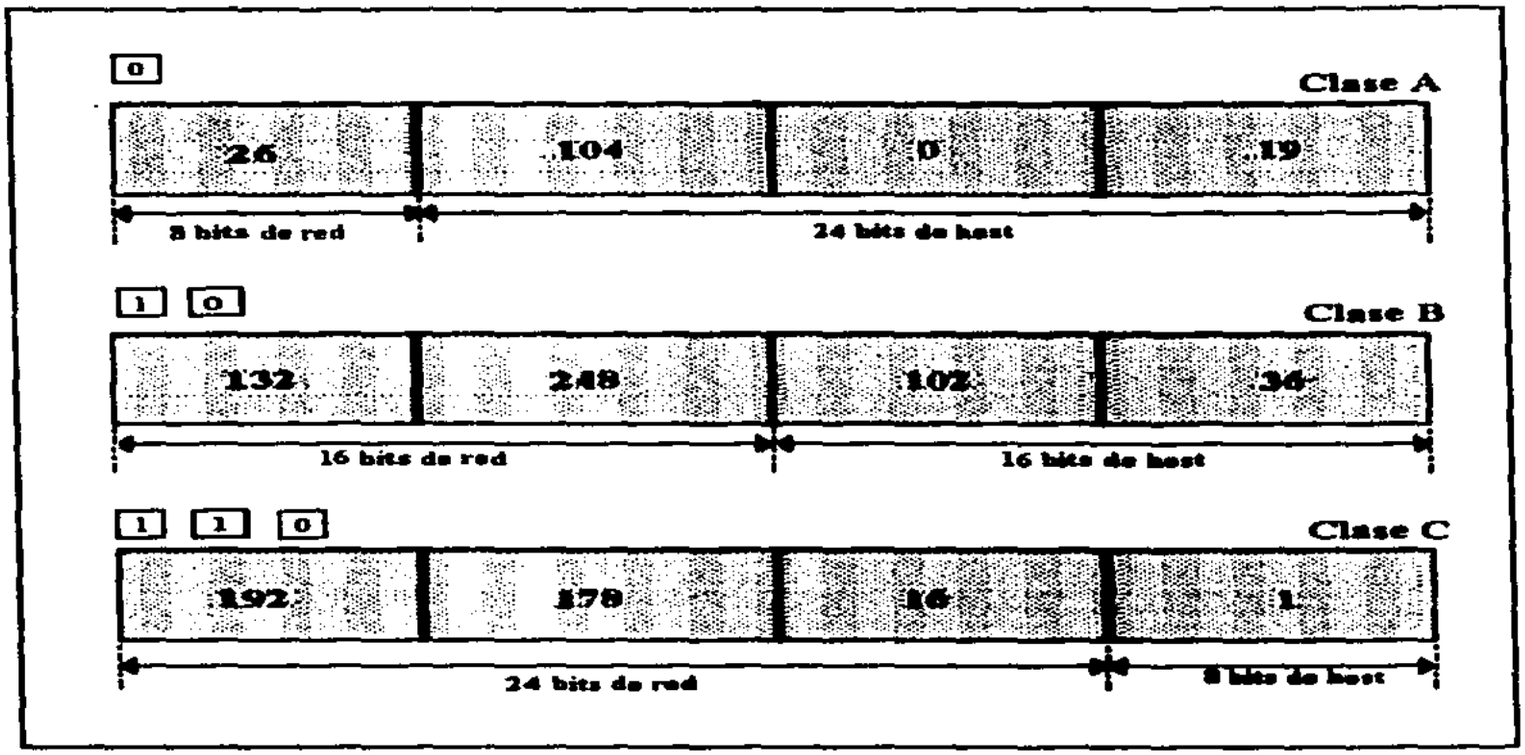

Figura 1.10. Estructura de Direcciones IP.

No todas las direcciones de la red o del host están disponibles para su uso. Ya hemos dicho que las direcciones cuyo primer byte sea mayor a 223 están reservadas. También hay dos direcciones de clase A, 0 y 127, que están reservadas para usos especiales. La red 0 está asignada a la ruta de default y la red 127 es la dirección de loopback. La ruta de default es utilizada para simplificar la información de ruteo que el IP debe manejar. La dirección de loopback simplifica las aplicaciones de la red permitiendo al host local ser direccionado de la misma forma que el host remoto. Utilizamos estas direcciones especiales de la red durante la configuración del host.

También hay algunas otras direcciones de host reservadas para usos especiales. En todas las clases de redes, los números de host 0 y 255 son reservados. Una dirección IP con todos los bits del host en cero identifican la red por si misma. Por ejemplo, 26.0.0.0, se refiere a la red 26 y 132.248.0.0 se refiere a la red 132.248. Direcciones en esta forma son utilizadas en las tablas de ruteo cuando se refieren a las redes por completo.

Una dirección IP con todos sus bits en uno es una dirección broadcast. Una dirección broadcast es utilizada para direccionar simultáneamente cada host de una red. La dirección broadcast de la red 132.248 es la 132.248.255.255. Un datagrama enviado a esta dirección es entregado a cada host individual de la red 132.248.

- Las direcciones IP son usualmente llamadas direcciones de host. Aunque este uso es muy común, es ligeramente equivocado. Las direcciones IP le son asignadas a las interfaces de la red, no a los sistemas de computadoras. Un gateway, como cuautitlanII, tiene una dirección diferente para cada red conectada a ésta. El gateway es conocido por otros dispositivos por medio de la dirección asociada con la red que comparte con dichos dispositivos. Por ejemplo, fesc direcciona a cuautitlanII como 132.248.102.254, mientras que la red U.N.A.M. lo direcciona como 132.248.229.253.

IP utiliza la parte de red de la dirección para rutear el datagrama entre las redes. La dirección completa, incluyendo la información del host, es utilizada para hacer la entrega final cuando el datagrama encuentra la red a la que estaba destinada.

1.3.2.1. Subredes

La estructura estándar de una dirección IP puede ser modificada localmente utilizando bits de dirección de host como bits de dirección de red adicionales. Esencialmente, la " línea divisoria " entre los bits de dirección de red y los de dirección de host es movida, creando redes adicionales, pero reduciendo el número máximo de hosts que pueden pertenecer a cada red. Estos bits recién asignados definen una red dentro de otra red más grande, llamada subred.

Una subred se define aplicándole un bit de máscara, la máscara de subred, a la dirección IP. Si un bit está encendido en la máscara, ese bit equivalente en la dirección es interpretado como un bit de red. Si un bit en la máscara está apagado, el bit pertenece a la parte de host de la dirección. La subred es conocida sólo localmente. Para el resto de la Internet, la dirección sigue siendo interpretada como una dirección IP estándar.

Muchos administradores de red prefieren utilizar las máscaras orientadas por medio de bytes porque son más fáciles de leer y entender. Sin embargo, definir las máscaras de subred en los límites de los bytes no es un requerimiento. La máscara de subred es orientada por medio de bits y puede ser aplicada a cualquier clase de dirección. Por ejemplo, una organización pequeña puede subdividir una dirección de clase C en cuatro subredes con la máscara 255.255.255.192. Aplicando esta máscara a una dirección de clase C se definen los dos bits más altos del cuarto byte como la parte subred de la dirección.

1.3.3. Tabla de Ruteo

Debido a que el ruteo está orientado por medio de la red, IP toma sus decisiones de ruteo basándose en la dirección de la parte de red. El módulo IP determina la parte de red de la dirección IP destino checando los bits más altos de la dirección para determinar la clase de la dirección. La clase de la dirección determina la porción de la dirección que IP utiliza para identificar la red. Si la red destino es la red local, la máscara de subred local es aplicada a la dirección destino.

Después de determinar el destino de la red, el módulo IP busca la red en la tabla de ruteo local. Los paquetes son ruteados hacia sus destinos tal y como son dirigidos por la tabla de ruteo. La tabla de ruteo puede ser elaborada por el administrador del sistema o por protocolos de ruteo, pero los resultados son los mismos; las decisiones de ruteo del IP son simples búsquedas a la tabla.

Se puede observar el contenido de la tabla de ruteo por medio del comando netstat -nr. La opción -r le dice a netstat que despliegue la tabla de ruteo y la opción -n le dice a netstat que despliegue la tabla en forma numérica. Es muy útil desplegar la tabla de ruteo en forma numérica ya que el destino de la mayoría de las rutas es una red y las redes están referidas generalmente por números de red.

El comando netstat muestra una tabla de ruteo conteniendo los siguientes campos para las versiones de UNIX como, UNIX SCO, SunOS, Solaris, HP-UX, y AIX:

Destination	La red (o host) destino.
Gateway	El gateway que se va a utilizar para buscar un destino específico.
Flags	Las banderas describen ciertas características de la ruta. Los posibles valores de las banderas son:
	U Indica que la ruta está activa y esta operando.
	H Indica que es una ruta específica hacia un host (muchas rutas van hacia las redes).

G Significa que la ruta utiliza un gateway. Las interfaces de red del sistema proveen rutas a redes directamente conectadas. Las demás rutas utilizan gateways remotos. Las redes directamente conectadas no manejan la bandera G; las demás si la utilizan.

D Significa que esta ruta fue añadida debido a una redirección ICMP. Cuando un sistema reconoce una ruta vía redirección ICMP, ésta añade la ruta a su tabla de ruteo, así, los paquetes adicionales enviados a ese destino ya no necesitan ser redireccionados. El sistema utiliza la bandera D para marcar esas rutas.

Ref Muestra el numero de veces que la ruta ha sido tomada como referencia para establecer una conexión.

Use Muestra el número de paquetes transmitidos por esta ruta.

Interface Nombre de la interface de red utilizada por esta ruta.

IRIX no tiene el campo Ref del informe de la tabla de ruteo pero contiene el campo MTU que muestra el valor de MTU colocado con el comando route para esa ruta, y los campos de RTT y RTTvar que muestran el tiempo del viaje redondo estimado (RTT) y la varianza en RTT para rutas con grandes cantidades de tráfico de TCP. Linux adiciona el campo Genmask que es la máscara de red de la ruta, y Metric la medida de costo para la ruta.

La primera entrada en la tabla es la ruta de loopback para el host local. Esta es la dirección de loopback mencionada anteriormente como un número reservado de la red. Debido a que cada sistema utiliza la ruta de loopback para enviar datagramas a sí mismo, esta entrada se encuentra en la tabla de ruteo de cada host. La bandera H está marcada porque es la ruta hacia un host específico (127.0.0.1), no la de la red entera (127.0 0.0).

Otra entrada única en la tabla de ruteo es la entrada con la palabra "default" en el campo Destination. Esta entrada es debida a la ruta por default, y el gateway específico para esta entrada es el gateway por default. El gateway por default es utilizado en el momento en que no hay una ruta específica en la tabla para una dirección de red destino.

Todos los gateways que aparecen en una tabla de ruteo están en redes directamente conectadas al sistema local. La figura 1.11 muestra cómo funciona el ruteo en una red. La capa IP de cada host y gateway es reemplazada por una pequeña pieza de la tabla de ruteo, mostrando las redes destino y los gateways utilizados para alcanzar esos destinos.

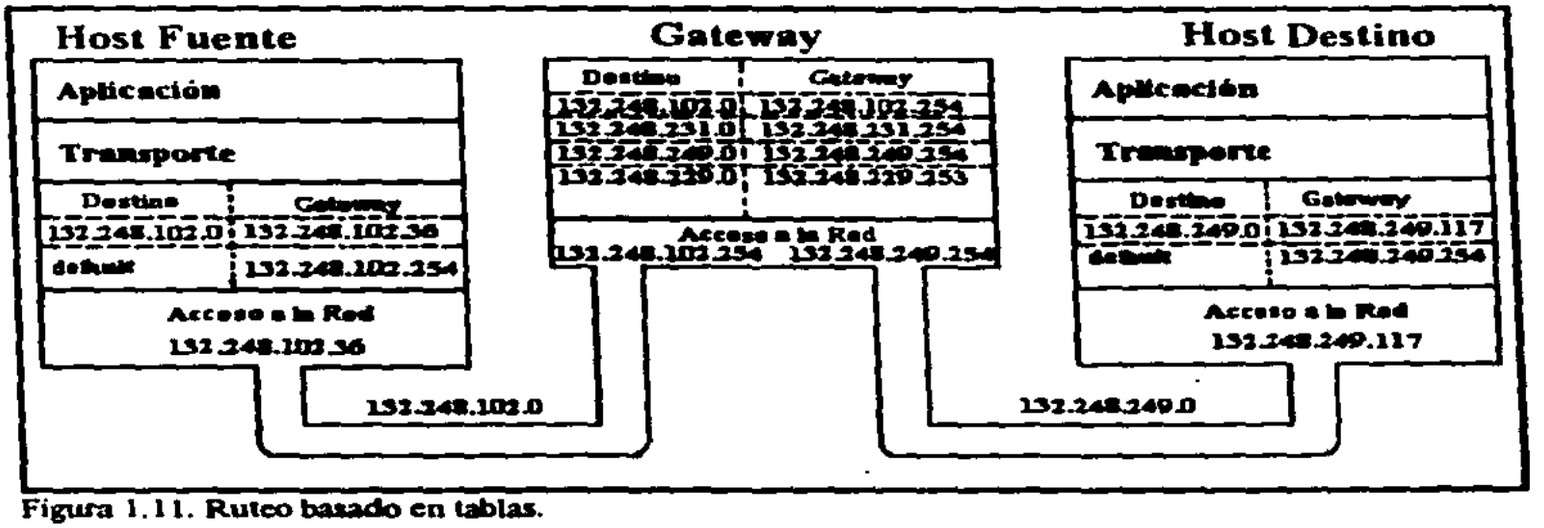

Figura 1.11. Ruteo basado en tablas.

Una tabla de ruteo no contiene rutas fin-a-fin. Una ruta sólo apunta al siguiente gateway, llamado el siguiente salto, a través de la trayectoria a la red destino. El host confía en el gateway local para enviar los

datos, y el gateway confía en otros gateways. Mientras un datagrama se mueve de un gateway a otro, en un momento llegará a uno que esté directamente conectado a esa red destino. Es este último gateway el que finalmente entregará los datos al host destino.

1.3.4. Determinacion de Direcciones.

La dirección IP y la tabla de ruteo dirigen el datagrama hacia una red física especifica, pero cuando los datos viajan a través de la red, deben obedecer los protocolos de la capa física utilizados por esa red. Las redes físicas que refuerzan a la red TCP/IP no entienden el direccionamiento IP. Las redes físicas tienen sus propios esquemas de direccionamiento, y existen tantos esquemas diferentes de direccionamiento como tantos tipos diferentes de redes físicas. Una tarea de los protocolos de acceso a la red es el mapear las direcciones IP hacia las direcciones de la red física.

El ejemplo más común de esta función de la capa de acceso a la red es el de la traducción de las direcciones IP a direcciones Ethernet. El protocolo que realiza esta función es el Address Resolution Protocol (ARP), el cual está definido en RFC 826.

El comando arp muestra el contenido de la tabla ARP. Para desplegar la tabla ARP completa utilice el comando arp -a. Las entradas individuales pueden ser desplegadas especificando el nombre del host en la linea del comando arp.

1.3.5. Internet Control Mensage Protocol

Una parte integral de IP es el Internet Control Mensage Protocol (ICMP) definido en RFC 792. Este protocolo es parte de la Capa de Internet y usa el datagrama IP para facilitar el envio de sus mensajes. ICMP envia mensajes que desempeñan el control de flujo, reporte de errores, e información funcional para TCP/IP:

Control de flujo	Cuando datagramas arriban demasiado rápido para procesarce, el host de destino o un gateway intermedio envía un Mensaje de Fuente Apagada ICMP detrás del envio. Esta dice a la fuente detener el envio temporalmente de datagramas.
Detectando destinos inalcanzables	Cuando un destino es inalcanzable, el sistema detecta el problema enviando un Mensaje de Destino Inalcanzable a la fuente del datagrama. Si el destino inalcanzable es una red o host, el mensaje es enviado por un gateway intermediario. Pero si el destino es un puerto inalcanzable, el host de destino envia el mensaje.
Redireccionando rutas	Una gateway envía un Mensaje de Redirección ICMP para decir que un host utilizara otro gateway, probablemente debido a que el otro gateway es una opción mejor. Este mensaje puede solamente ser utilizado cuando el host fuente está en la misma red como ambos gateways.
Checando remotos hosts	Un host pueden enviar un Mensaje Eco ICMP para ver si el Internet Protocol del sistema remoto está activo y operacional. Cuando un sistema recibe un mensaje de eco, envía el mismo paquete después al host fuente.

1.4. Capa de Transporte

La capa de protocolo justamente sobre la Capa de Internet es la Capa de Transporte host-a-host. Este nombre es reducido generalmente a Capa de Transporte. Los dos protocolos más importantes en la Capa de Transporte son Transmission Control Protocol (TCP) y User Datagram Protocol (UDP). TCP provee servicio de entrega de datos confiable con detección de error punto-a-punto y corrección. UDP provee servicio de entrega de datagramas sin conexión. Ambos protocolos suministran información entre la Capa de Aplicación y la Capa de Internet. Los programadores de aplicaciones pueden escoger cualquier servicio más apropiado para sus aplicaciones específicas.

1.4.1. User Datagram Protocol

El User Datagram Protocol da acceso directo a programas de aplicación para el servicio de entrega de datagramas, ligado al servicio de entrega que IP provee. Esto permite a aplicaciones intercambiar mensajes sobre la red con un mínimo protocolo.

UDP es un protocolo sin conexión de datagramas. (Como se notó anteriormente, " sin conexión " significa que simplemente que no hay técnicas en el protocolo para verificar que la información alcanzó la otra terminal de la red correctamente.). Dentro de la computadora, UDP suministrará información correctamente. UDP utiliza 16 bits para los números Puerto Fuente y Puerto Destino en la palabra 1 del encabezado de mensaje, para suministrar datos a los correctos procesos de aplicación. La figura 1.12 muestra el formato de mensaje de UDP.

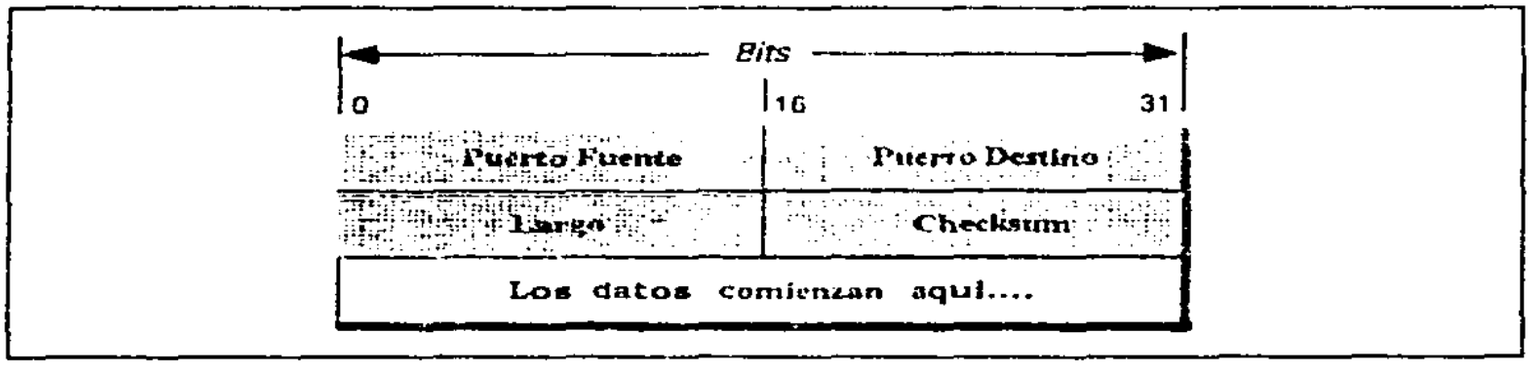

Figura 1.12. Formato de un Mensaje UDP.

Si la cantidad de información que se transmite es pequeña, por encima de crear conexiones y asegurar la entrega confiable puede ser mayor que el trabajo de retransmitir el conjunto de datos completo. En este caso, UDP es la opción más eficiente para un protocolo de Capa de Transporte. Las aplicaciones que se ajustan al modelo " consulta-respuesta " son también excelentes candidatos para utilizar UDP. La respuesta puede ser utilizada como un reconocimiento positivo para la consulta. Si una respuesta no es recibida dentro de un cierto período, la aplicación envía justamente otra consulta. Aún otras aplicaciones proveen sus propias técnicas para entrega de datos confiable, y no requiere ese servicio de protocolo de la capa de transporte. El imponer otra capa de reconocimiento en cualesquier de estos tipos de aplicaciones es ineficiente.

1.4.2. Transmission Control Protocol

Las aplicaciones que requieren el protocolo de transporte para proveer de entrega de datos confiable usan TCP porque verifican los datos exactamente que se entregan a través de la red y en la secuencia adecuada. TCP es un protocolo orientado a conexión, byte-stream. Vamos a mirar cada uno de los términos anteriores, orientado a conexión, y byte-stream en detalle.

TCP provee confiabilidad con un mecanismo llamado Reconocimiento Positivo con Retransmisión (PAR). Simplemente planteado, un sistema utilizando PAR envía los datos de nuevo, a menos que reciba noticias del sistema remoto que los datos han llegado bien. La unidad de datos intercambiada entre módulos cooperativos TCP es llamado un segmento (ver Figura 1.13). Cada segmento contiene un checksum este recipiente es usado para verificar que el dato este intacto. Si el segmento de dato es recibido intacto, el receptor envía un Reconocimiento Positivo después al que envío. Si el segmento de dato está dañado, los descarta el receptor. Después de un periodo de interrupción temporal apropiado, el módulo de envío TCP retransmite cualquier segmento para el que ningún reconocimiento positivo ha estado recibiendo.

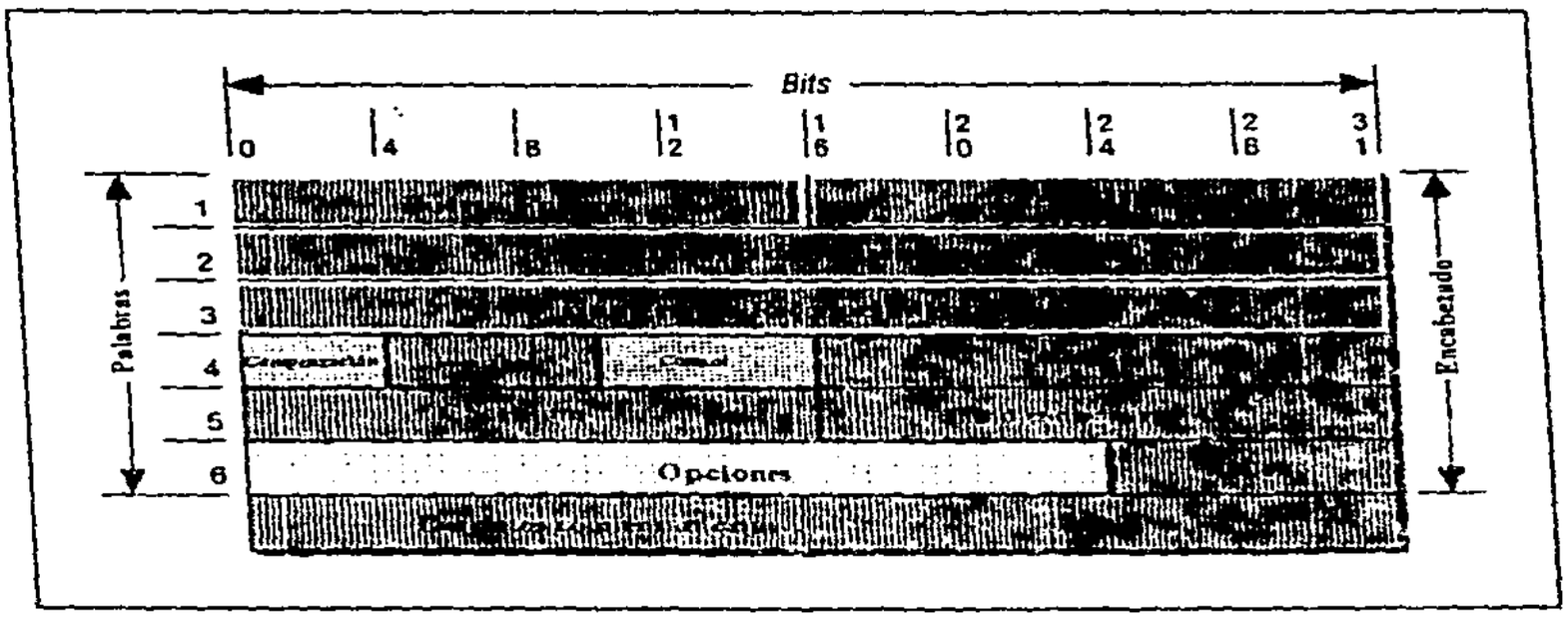

Figura 1.13. Formato de un Segmento TCP.

TCP es orientado a conexión. Establece una conexión lógica punto-a-punto entre los dos host comunicados. Información de control, llamado un handshake, es intercambiada entre los dos puntos finales para establecer un diálogo antes de que los datos sean transmitidos. TCP indica la función de control en un segmento colocando el apropiado bit en el campo de Control en la palabra 4 del encabezado de segmento.

El tipo de handshake utilizado por TCP es llamado un *three-way handshake* porque tres segmentos son intercambiados. La figura 1.14 muestra la forma más simple de un three-way handshake.

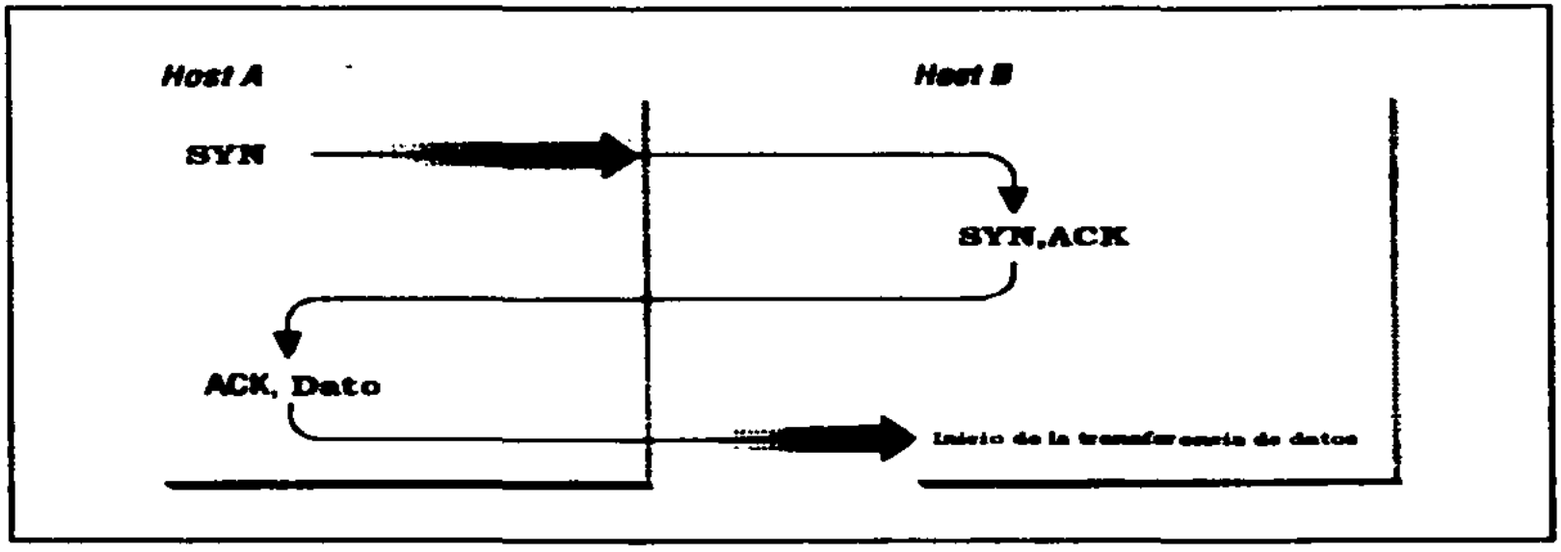

Figura 1.14. Three-way Handshake.

Después de este intercambio, el host A que utiliza TCP tiene evidencia positiva que el TCP remoto esta activo y listo para recibir datos. Tan pronto como la conexión es establecida, los datos pueden ser transferidos. Cuando los módulos cooperativos han concluido las transferencias de datos, intercambiará un three-way handshake con segmentos conteniendo el bit " No más datos por enviar " (llamado el bit FIN) cerrando la conexión. Este es el intercambio punto-a-punto de datos que provee la conexión lógica entre los dos sistemas.

TCP visualiza los datos enviados como una corriente continua de bytes, no como paquetes independientes. Por lo tanto, TCP tiene cuidado por mantener la secuencia de que bytes son enviados y recibidos. Los campos "Número de Secuencia" y "Número de Reconocimiento" en el encabezado del segmento TCP siguen el rastro de los bytes.

Para seguir el rastro del stream de datos correctamente, cada fin de la conexión tiene que conocer el número inicial de la otra terminal final. Los dos extremos de la conexión sincronizan sistemas de numeración de byte para intercambiar segmentos SYN durante el handshake. El campo " Número de Secuencia " en el segmento SYN contiene el Número Inicial de Secuencia (ISN), que es el punto de partida para el sistema de numeración de byte. Aunque no es requerido por el protocolo estándar, el ISN es generalmente 0.

El Número de Secuencia en el encabezado de un segmento de datos identifica la posición secuencial en el stream de datos del primer byte de información en el segmento.

El Segmento de Confirmación (ACK) desempeña dos funciones un reconocimiento positivo y el control de flujo. La confirmación le dice al que envía cuantos datos han sido recibidos, y cuánto más el receptor puede aceptar. El Número de Reconocimiento es el número de secuencia del byte último recibido en el fin remoto.

El campo de Ventana contiene la cantidad de bytes que el fin remoto es capaz de aceptar. Si el receptor es capaz de aceptar más de 6000 bytes, la Ventana debería ser 6000 La ventana indica al transmisor que puede continuar enviando segmentos en tanto que el número total de bytes que envía es más pequeño que la ventana de bytes que el receptor pueda aceptar. El receptor controla el flujo de bytes del transmisor cambiando el tamaño de la ventana. Una ventana cero dice al transmisor cesar la transmisión hasta que reciba un valor no cero de ventana.

La figura 1.15 muestra un stream de datos de TCP que comienza con un Número Inicial de Secuencia 0. El sistema receptor ha recibido y reconocido 2000 bytes, de modo que el Número de Reconocimiento es 2000. El receptor tiene también espacio de buffer suficiente para otros 6000 bytes, de modo que ha anunciado una Ventana de 6000. El transmisor está enviando actualmente un segmento de inicio de 1000 bytes con Número de Secuencia 4001. El transmisor no ha recibido un reconocimiento para los bytes desde el 2001, pero continúan enviando datos en tanto que se este dentro del tamaño de la ventana. Si el transmisor llena la ventana y no recibe un reconocimiento de la información previamente enviada, después de una interrupción temporal apropiada, envía la información otra vez a partir de el primero byte sin reconocer. En la Figura 1.15, la retransmisión comenzaría desde el byte 2001 si ningún posterior reconocimientos se hubiera recibido. Este procedimiento asegura que esos datos son recibidos confiadamente en el extremo más alejado de la red.

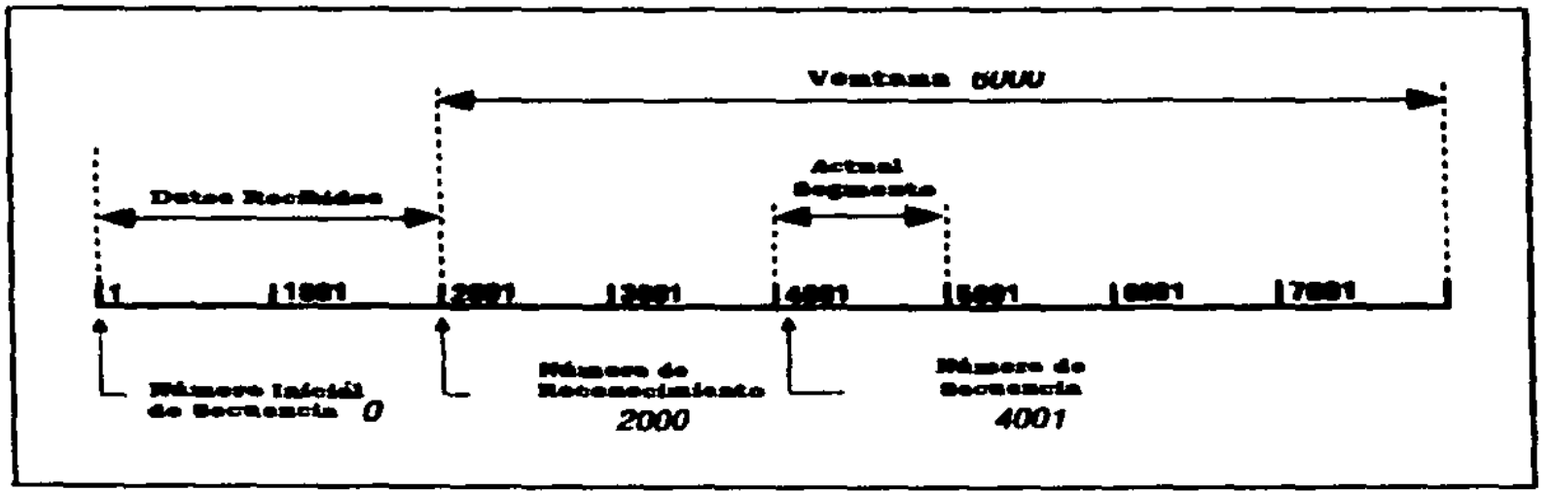

Figura 1.15. Flujo de Datos TCP.

TCP es también responsable por entregar datos recibidos de IP a la aplicación correcta. Los datos que pertenecen a una aplicación están identificados por un número de 16 bits llamada el número de puerto. El Puerto Fuente y el Puerto de Destino están contenidos en la primera palabra del encabezado del segmento. Pasando correctamente datos desde y hacia la Capa de Aplicación es una parte importante de la cual servicios de Capa de Transporte hacen uso.

1.5. Protocolos, Puertos y Sockets

Una vez que los datos son ruteados a través de la red y entregados al host específico, éstos deben ser entregados al usuario o proceso correctos. Como los datos se mueven hacia arriba y abajo en las capas del TCP / IP, es necesario un mecanismo para enviar los datos hacia los protocolos correctos de cada capa. El sistema debe ser capaz de combinar datos desde varias aplicaciones hacia pocos protocolos de transporte, y desde protocolos de transporte hacia el Internet Protocol. Para completar este proceso, IP utiliza números de protocolos para identificar los protocolos de transporte, y los protocolos de transporte utilizan números de puertos para identificar las aplicaciones. Algunos números de protocolos y puertos están reservados para identificar los servicios estándares.

1.5.1. Números de Protocolos.

El número de protocolo es un único byte en la tercera palabra del encabezado del datagrama. El valor identifica el protocolo en la capa de arriba de la capa IP hacia donde los datos deben ser enviados.

En un sistema UNIX los números de protocolos están definidos en /etc/protocols. Este archivo es una tabla simple que contiene el nombre del protocolo y el número de protocolo asociado con este nombre. El formato de la tabla es de una entrada por línea, consistente del nombre oficial del protocolo, separado por un espacio en blanco de su número de protocolo. El número del potocolo está separado por un espacio en blanco del "alias" perteneciente al nombre del protocolo. Los comentarios en la tabla comienzan con un #.

nombre_del_protocolo número_de_protocolo alias #comentario

1.5.2. Números de Puertos

Después de pasar IP los datos entrantes hacia el protocolo de transporte, el protocolo de transporte pasa los datos hacia los procesos de aplicación correctos. Los procesos de aplicación (también llamado servicios de red) están identificados por números de puertos, los cuales son valores de 16 bits. El "número de puerto fuente, que identifica al proceso que envió los datos, y el "número de puerto destino", que identifica el proceso que va a recibir los datos están contenidos en la primera palabra del encabezado de cada segmento TCP y paquete UDP. En sistemas UNIX, los números de puertos están definidos en el archivo /etc/services.

El formato de este archivo es muy similar al de /etc/protocols. Cada línea comienza con el nombre oficial del servicio, separado por un espacio en blanco del número de puerto / protocolo asociados con tal servicio. Los números de puertos están en par con los nombres del protocolo de transporte, ya que diferentes protocolos de transporte pueden utilizar el mismo número de puerto. Una lista opcional de alias para el nombre del servicio oficial pueden aparecer después del par número de puerto/protocolo.

nombre_del_servicio número_puerto/protocolo alias #comentario

Esta tabla, combinada con la tabla /etc/protocols, proveen toda la información necesaria para enviar los datos hacia la aplicación correcta. Un datagrama alcanza su destino basado en la dirección destino que se encuentra en la quinta palabra en el encabezado del datagrama. IP utiliza el número de protocolo en la tercera palabra en el encabezado del datagrama, para enviar los datos del datagrama, al protocolo de capa de transporte correcto. La primera palabra de los datos enviados al protocolo de transporte contiene el número de puerto destino el cual le dice al protocolo de transporte que suba los datos hacia la aplicación especifica. La figura 1.16 muestra este proceso de envío.

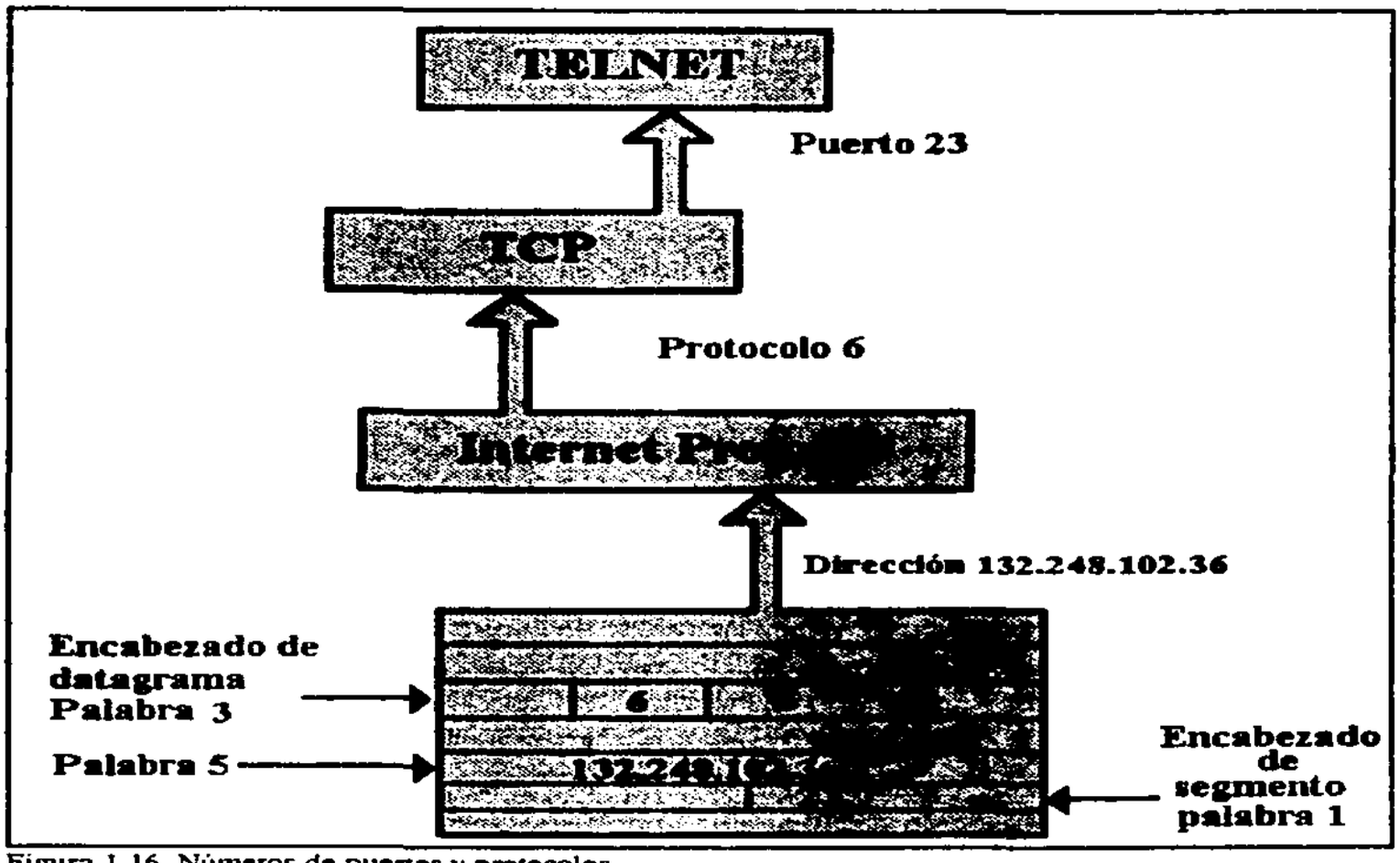

Figura 1.16. Números de puertos y protocolos

1.5.3. Sockets

Existe un segundo tipo de número de puerto llamado puerto asignado dinámicamente. Como el nombre implica, los puertos asignados dinámicamente no son pre-asignados. Estos son asignados a los procesos cuando son necesitados. El sistema se asegura de no asignar el mismo número de puerto a dos procesos, y que el número asignado esté por arriba del rango del número de puertos estándar.

Los puertos asignados dinámicamente proveen de la flexibilidad necesaria para soportar múltiples usuarios. Si a un usuario de TELNET se le asigna el puerto número 23 para ambos puertos fuente y destino; ¿qué números de puertos son asignados al siguiente usuario de TELNET? Para identificar específicamente a cada conexión, al puerto fuente se le asigna un número de puerto dinámicamente, y el número de puerto conocido se le utiliza como el puerto destino.

Este es el par de números de puertos, fuente y destino, que identifican exclusivamente a cada conexión de la red. El host destino conoce el puerto del fuente, porque éste se encuentra en el encabezado del segmento TCP y en el encabezado del paquete UDP. Ambos hosts conocen el puerto destino porque es un puerto conocido. La figura 1.17 muestra el intercambio de números de puertos durante el handshake TCP.

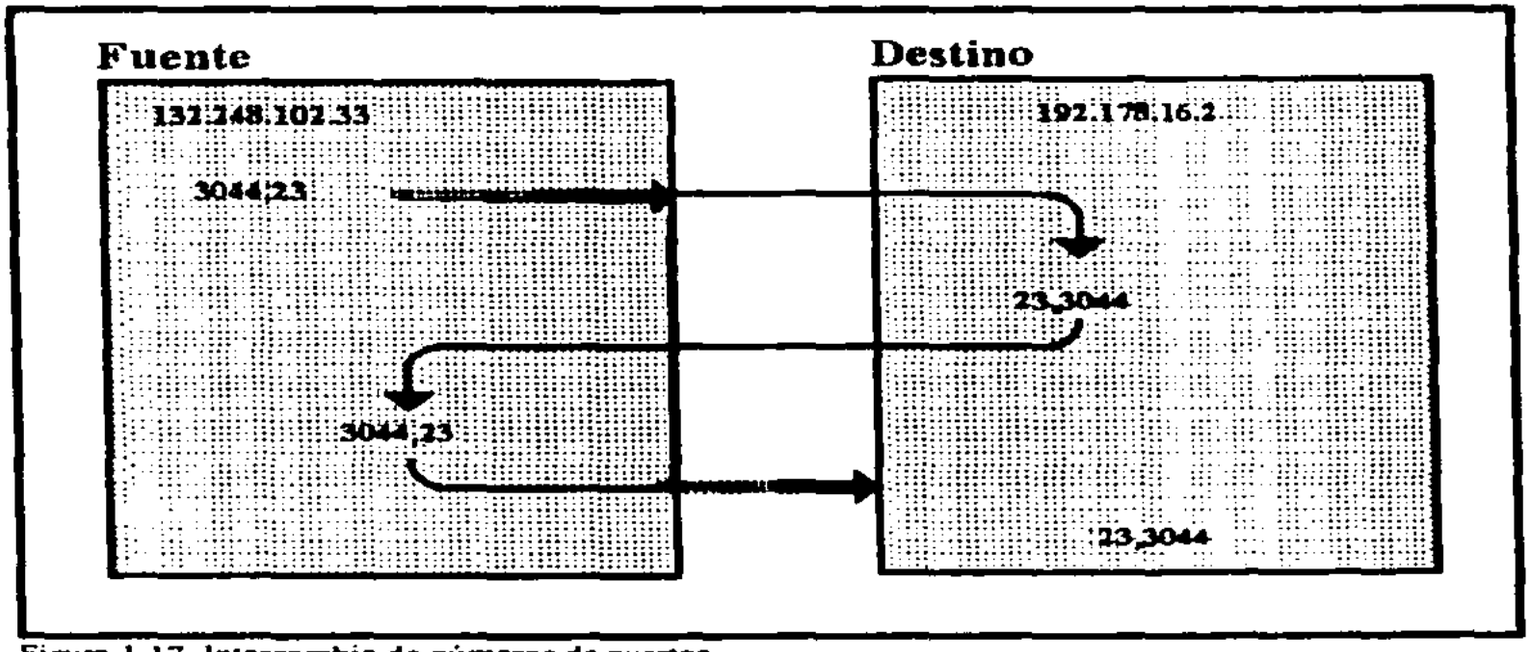

Figura 1.17. Intercambio de números de puertos.

A la combinación de una dirección IP y un número de puerto se le llama socket. Un socket únicamente identifica un proceso de red único dentro de toda la Internet. Algunas veces los términos "socket" y "número de puerto" son utilizados de forma intercambiable. De hecho, a los servicios conocidos frecuentemente se les llama "sockets conocidos". En el contexto de esta discusión, un "socket" es la combinación de una dirección IP y un número de puerto. Un par de sockets, uno para el host que recibe y uno para el host que envía, define la conexión para protocolos orientados a conexión tal como el TCP.

EL puerto del socket destino es conocido por ambos sistemas ya que éste es un puerto conocido. El puerto del socket de la fuente es conocido, porque el host origen informó al host destino acerca del socket de la fuente cuando la petición de conexión fue hecha. El par de sockets es conocido por ambas computadoras la de origen y la de destino. La combinación de los dos sockets identifica esta conexión única; no existe otra conexión en la Internet que cuente con este par.

1.6. Capa de Aplicación

En la parte superior de la arquitectura de protocolo de TCP/IP es la Capa de Aplicación. Esta capa incluye todos los procesos que utilizan los protocolos de Capa de Transporte para suministrar información. Hay muchos protocolos de aplicaciones. Muchos proveen servicios al usuario, y nuevos servicios se están siempre adicionando a esta capa.

La figura 1.18 muestra la jerarquía de protocolos en una computadora imaginaria. Cuando mire esta figura, por favor recuerde que reducir la complejidad de un amontonó de protocolos a un diagrama de bloques es, por su propia naturaleza, una simplificación. Esta ilustración es solamente para ayudar a visualizar la relación de los muchos protocolos en un host único. No todos los protocolos mostrados en Figura 1.14 han sido discutidos todavía, pero debería ser de gran ayuda obtener una idea de la estructura global. En la parte superior de la figura están los protocolos de aplicaciones, como FTP Y TELNET. Cada protocolo está mostrado con el número del RFC que lo define. La línea corre desde cada caja al servicio de capa más bajo que el protocolo utiliza. Vemos que FTP, TELNET, y SMTP confían principalmente en TCP; mientras NFS, DNS, y RIP confían principalmente en UDP. Algunas aplicaciones tipo protocolo, como el Exterior Gateway

Protocol (EGP), otro protocolo de ruteo, no utiliza servicios de Capa de Transporte; utilizan servicios de IP directamente.

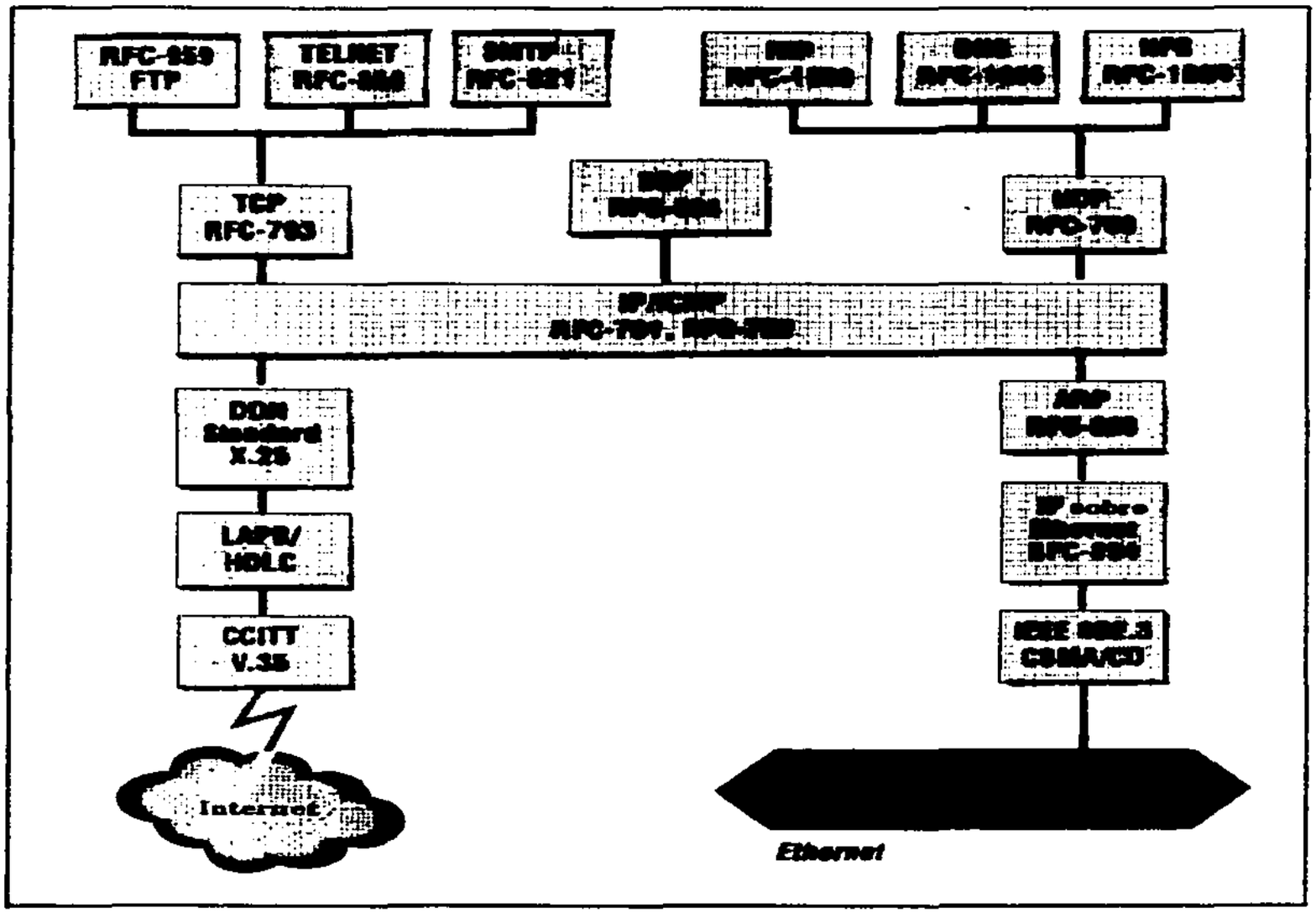

Figura 1.18. Protocolos TCP / IP Dentro de un Sinple Gateway.

Debajo de las aplicaciones son los protocolos de Capa de Transporte: TCP Y UDP. Ellas intercambian directamente con IP. Todos los datos, dentro y fuera del sistema, fluyen a través de IP. IP suministra datos de las capas superiores a la red correcta, y suministra datos de la red al servicio de transporte correcto. Así mismo, los servicios de transporte suministran los datos que reciben de IP a la aplicación correcta.

Amontonamos todos los protocolos debajo de IP juntos como protocolos de Acceso a la Red. El apilamiento de estos protocolos en tres capas no está pretendido para implicar alguna otra jerarquía, aunque en el lado X.25 estos protocolos corresponden a los tres capas más bajas de la jerarquía de OSI (Red, Enlace de Datos, y Físico). Para TCP / IP, todos estos protocolos son protocolos de Acceso a la Red.

CAPITULO 2 Proceso de Inicialización y Guiones de Inicio de la Red en UNIX

Comprender los guiones de inicialización del sistema es una parte vital de administración del sistema. Debemos saber donde están localizados y lo que ellos hacen. De esa manera, se puede poder reconocer cualquier problema en tiempo de arranque, y conocer qué acción correctiva hay que tomar. También, de vez en cuando probablemente haya necesidad de modificarlos para añadir nuevos servicios (o para desactivar unos que se han decidido no necesitar).

Aunque los nombres, ubicaciones de directorio, y el código de programa shell para guiones de inicialización del sistema varían ampliamente entre versiones basadas en BSD de UNIX y aquellas derivadas de Sistema V, las actividades logradas por cada conjunto de guiones difieren como un todo solamente en modos menores. En términos de alto nivel, el proceso de arranque BSD está controlado por un número pequeño de guiones (2-5) en el directorio /etc teniendo comienzo de nombres con o terminación en rc, que están ejecutándose generalmente secuencialmente. En contraste, Sistema V ejecuta un gran número de guiones cuando cambia de nivel de arranque, organizados en unas tres jerarquías, utilizando separados guiones rcn y directorios rcn.d. Los archivos en el directorio rcn.d son vinculados generalmente a aquellos archivos en el directorio init.d, donde los archivos reales se encuentran, y la porción final de el nombre de archivo en el directorio rcn.d es el mismo que el nombre de fichero en el directorio init.d.

En general, init controla el proceso de arranque en modo de multiusuario. init corre cualquier guión de inicialización que ha sido diseñado para correr, y la estructura del programa init determina el diseño fundamental del conjunto de guiones de inicialización para esa versión de UNIX: Cual de los guiones es nombrado, donde están localizado en el sistema de archivo, la secuencia en que son ejecutados, las restricciones hechas por los programadores de guiones y las suposiciones bajo que operan, y así sucesivamente. Finalmente, es init la diferencia en las versiones de Sistema V y el BSD que determinan las diferencias en el proceso de arranque para los dos tipos de sistemas.

En la mayoría de los sistemas, subsistemas puramente locales que no utilizan o de cualquier modo dependen de la red son arrancados generalmente antes de que la red sea inicializada, y aquellos subsistemas que necesitan facilidades de red son comenzados después de la red.

La inicialización de la red comienza por colocar el nombre del host del sistema en la red si es necesario y configurando las interfaces de red, posibilitándola para comunicarce con otros hosts en la red local, así como con otros sistemas más allá asequibles. Rutas estáticas pueden también ser definidas en este punto utilizando el comando route.

Los servicios de red también cuentan con varios procesos demonio. También están inicializados generalmente por comandos de la forma siguiente:

```
if [ -x ruta_del_servidor ]; then
        comandos preparatorios
        comando_que_inicializa_el_servidor
        echo Starting nombre_del_servidor
fi
```

Cuando el archivo del programa servidor existe y es ejecutable, el guión desempeña cualquier actividad preparatoria necesaria y entonces inicia el proceso servidor. Observe que algunos servicios se ejecutan automáticamente mientras que otros tienen que ser inicializados explícitamente.

Una vez corriendo la red básica, otros servicios y subsistemas que dependen de ellos puede ser inicializados.

Las siguientes tablas muestran los archivos de configuración de red para TCP/IP utilizados en las diferentes versiones de UNIX y la función que realiza cada uno de ellos.

Versión de UNIX	Archivos de configuración de red	
SunOS	/etc/rc.boot	Coloca el nombre del host del sistema en la red (hostname).
		El programa ifconfig da a conocer las interfaces de red, como el loopback y las tarjetas EtherNet, al kernel de UNIX; esto activa las interfaces de red especificadas y les asigna una dirección IP a cada una.
		El programa route maneja la tabla de ruteo del núcleo y sirve para establecer rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
	/etc/rc.local	El comando domainname pone el nombre de dominio NIS, algunos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS.
		ifconfig reestablece la direccion broadcast y la máscara de subred para cada interface de red.
		El programa route reestablece rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
		Se inicia el demonio route (routed) que mantiene información dinámica acerca de las redes con que nuestro sistema se comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.
		El servidor de nombres es activado (named), que provee nombres de hosts remotos para TCP/IP dinámicamente.
		Se inicia el demonio sendmail que manipula el correo electrónico entre sistemas.
	/etc/rc	Se inicia el demonio inetd que es el servidor maestro de la red responsable por coordinar muchos tipos de solicitudes de red a través de un gran número de demonios subordinados que controlan y delegan tareas; los servicios manegados por inetd son listados en el archivo /etc/inetd.conf, e incluye soporte para los comandos telnet, ftp, rlogin, rsh, rcp y finger.

Versión de UNIX	Archivos de configuración de red	
Solaris	/etc/init.d/rootusr	Coloca el nombre del host del sistema en la red (hostname).
		El programa ifconfig da a conocer las interfaces de red, como el loopback y las tarjetas EtherNet, al kernel de UNIX; esto activa las interfaces de red especificadas y les asigna una dirección IP a cada una.
		El programa route maneja la tabla de ruteo del núcleo y sirve para establecer rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
	/etc/init.d/inetinit	El comando domainname pone el nombre de dominio NIS, algunos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS.
		El programa route reestablece rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
		Se inicia el demonio route (routed) que mantiene información dinámica acerca de las redes con que nuestro sistema se

		comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.
	/etc/init.d/inetsvc	ifconfig reestablece la direccion broadcast y la máscara de subred para cada interface de red.
		El servidor de nombres es activado (named), que provee nombres de hosts remotos para TCP/IP dinámicamente.
		Se inicia el demonio inetd que es el servidor maestro de la red responsable por coordinar muchos tipos de solicitudes de red a través de un gran número de demonios subordinados que controlan y delegan tareas; los servicios manegados por inetd son listados en el archivo /etc/inetd.conf, e incluye soporte para los comandos telnet, ftp, rlogin, rsh, rcp y finger.
	/etc/init.d/sendmail	Se inicia el demonio sendmail que manipula el correo electrónico entre sistemas.

Versión de UNIX	Archivos de configuración de red	
HP-UX	/etc/rc	Coloca el nombre del host del sistema en la red (hostname).
	/etc/netlinkrc	El programa ifconfig da a conocer las interfaces de red, como el loopback y las tarjetas EtherNet, al kernel de UNIX; esto activa las interfaces de red especificadas y les asigna una dirección IP a cada una; ifconfig establece la direccion broadcast y la máscara de subred para cada interface de red.
		El programa route maneja la tabla de ruteo del núcleo y sirve para establecer rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
		Se inicia el demonio inetd que es el servidor maestro de la red responsable por coordinar muchos tipos de solicitudes de red a través de un gran número de demonios subordinados que controlan y delegan tareas; los servicios manegados por inetd son listados en el archivo /etc/inetd.conf, e incluye soporte para los comandos telnet, ftp, rlogin, rsh, rcp y finger.
	/etc/netnfsrc	El comando domainname pone el nombre de dominio NIS, algunos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS.
	/etc/netbsdsrc	Se inicia el demonio de ruteo gateway (gated) que mantiene información dinámica acerca de las redes con que nuestro sistema se comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.
		El servidor de nombres es activado (named), que provee nombres de hosts remotos para TCP/IP dinámicamente.
		Se inicia el demonio sendmail que manipula el correo electrónico entre sistemas.

Versión de UNIX	Archivos de configuración de red	
Linux	/etc/rc.d/rc.inet1	Coloca el nombre del host del sistema en la red (hostname).
		El programa ifconfig da a conocer las interfaces de red, como el loopback y las tarjetas EtherNet, al kernel de UNIX; esto activa las interfaces de red especificadas y les asigna una dirección IP a cada una; ifconfig establece la direccion broadcast y la máscara de subred para cada interface de red.
		El programa route maneja la tabla de ruteo del núcleo y sirve para establecer rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
	/etc/rc.d/rc.inet2	Se inicia el demonio inetd que es el servidor maestro de la red responsable por coordinar muchos tipos de solicitudes de red a través de un gran número de demonios subordinados que controlan y delegan tareas; los servicios manegados por inetd son listados en el archivo /etc/inetd.conf, e incluye soporte para los comandos telnet, ftp, rlogin, rsh, rcp y finger.
		El servidor de nombres es activado (named), que provee nombres de hosts remotos para TCP/IP dinámicamente.
		Se inicia el demonio route (routed) que mantiene información dinámica acerca de las redes con que nuestro sistema se comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.
	/etc/rc.d/rc.local	El comando domainname pone el nombre de dominio NIS, algunos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS.

Versión de UNIX	Archivos de configuración de red	
AIX	/etc/rc.net	Coloca el nombre del host del sistema en la red (hostname).
		El programa ifconfig da a conocer las interfaces de red, como el loopback y las tarjetas EtherNet, al kernel de UNIX; esto activa las interfaces de red especificadas y les asigna una dirección IP a cada una; ifconfig establece la direccion broadcast y la máscara de subred para cada interface de red.
		El programa route maneja la tabla de ruteo del núcleo y sirve para establecer rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
	/etc/rc.tcpip	Se inicia el demonio inetd que es el servidor maestro de la red responsable por coordinar muchos tipos de solicitudes de red a través de un gran número de demonios subordinados que controlan y delegan tareas; los servicios manegados por inetd son listados en el archivo /etc/inetd.conf, e incluye soporte para los comandos telnet, ftp, rlogin, rsh, rcp y finger.
		El servidor de nombres es activado (named), que provee nombres de hosts remotos para TCP/IP dinámicamente.
		Se inicia el demonio route (routed) que mantiene información dinámica acerca de las redes con que nuestro sistema se comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.

		Se inicia el demonio sendmail que manipula el correo electrónico entre sistemas.
	/etc/rc.nfs	El comando domainname pone el nombre de dominio NIS, algunos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS.

Versión de UNIX	Archivos de configuración de red	
IRIX	/etc/init.d/network	Coloca el nombre del host del sistema en la red (hostname).
		El programa ifconfig da a conocer las interfaces de red, como el loopback y las tarjetas EtherNet, al kernel de UNIX; esto activa las interfaces de red especificadas y les asigna una dirección IP a cada una; ifconfig establece la direccion broadcast y la máscara de subred para cada interface de red.
		El programa route maneja la tabla de ruteo del núcleo y sirve para establecer rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
		Se inicia el demonio route (routed) que mantiene información dinámica acerca de las redes con que nuestro sistema se comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.
		El servidor de nombres es activado (named), que provee nombres de hosts remotos para TCP/IP dinámicamente.
		El comando domainname pone el nombre de dominio NIS, algunos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS.
		Se inicia el demonio inetd que es el servidor maestro de la red responsable por coordinar muchos tipos de solicitudes de red a través de un gran número de demonios subordinados que controlan y delegan tareas; los servicios manegados por inetd son listados en el archivo /etc/inetd.conf, e incluye soporte para los comandos telnet, ftp, rlogin, rsh, rcp y finger.
		Se inicia el demonio sendmail que manipula el correo electrónico entre sistemas.

Versión de UNIX	Archivos de configuración de red	
SCO UNIX	/etc/tcp	Coloca el nombre del host del sistema en la red (hostname).
		El programa ifconfig da a conocer las interfaces de red, como el loopback y las tarjetas EtherNet, al kernel de UNIX; esto activa las interfaces de red especificadas y les asigna una dirección IP a cada una; ifconfig establece la direccion broadcast y la máscara de subred para cada interface de red.
		El programa route maneja la tabla de ruteo del núcleo y sirve para establecer rutas estáticas para otras computadoras o redes por medio de interfaces que ifconfig ha configurado y activado.
		Se inicia el demonio route (routed) que mantiene información dinámica acerca de las redes con que nuestro sistema se comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.

		El servidor de nombres es activado (named), que provee nombres de hosts remotos para TCP/IP dinámicamente.
		Se inicia el demonio inetd que es el servidor maestro de la red responsable por coordinar muchos tipos de solicitudes de red a través de un gran número de demonios subordinados que controlan y delegan tareas; los servicios manegados por inetd son listados en el archivo /etc/inetd.conf, e incluye soporte para los comandos telnet, ftp, rlogin, rsh, rcp y finger.
		Se inicia el demonio route (routed) que mantiene información dinámica acerca de las redes con que nuestro sistema se comunica, usado para determinar las rutas para transferir datos; este demonio es designado para remplazar una tabla de ruteo estatica construida por un número de comandos route durante el proceso de arranque.
		Se inicia el demonio sendmail que manipula el correo electrónico entre sistemas.
	/etc/default/tcp	El comando domainname pone el nombre de dominio NIS, algunos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS.

2.1. Demonio Internet

Algunos demonios de red que sirven a algunos protocolos, son explícitamente iniciados incluyéndolos en los archivos de arranque.Muchos otros demonios de red no son iniciados individualmente. Esos demonios son iniciados por un super servidor que escucha las peticiones de servicios de la red e inicia el demonio apropiado para procesar la petición. Este super servidor es llamado el demonio internet.

El demonio internet (inetd) es iniciado al tiempo de arranque de un archivo de inicialización como /etc/rc para SunOS, /etc/init.d/inetsvc para Solaris, /etc/netlinkrc para HP-UX, /etc/rc.d/rc.inet2 para Linux, /etc/rc.tcp para AIX, /etc/init.d/network para IRIX, y /etc/tcp para UNIX SCO. Cuando es iniciado, inetd lee su configuración del archivo /etc/inetd.conf. Este archivo contiene los nombres de los servicios que inetd espera recibir e iniciar. Podemos añadir o eliminar servicios haciendo cambios al archivo inetd.conf.

Los campos en la entrada inetd.conf son, de izquierda a derecha:

name type protocol wait-status uid server arguments

name — El nombre de un servicio, como es listado en el archivo /etc/services.

type — El tipo de servicio de entrega de datos utilizado, también llamado tipo de socket. Los tipos de socket más utilizados comúnmente son:

- stream — El servicio de entrega stream provisto por TCP, por ejemplo, TCP byte stream.
- dgram — El servicio de entrega de paquetes (datagramas), provisto por UDP.
- raw. — Servicio directo de datagramas IP

protocol — Este es el nombre de un protocolo, como es dado en el archivo /etc/protocols. Su valor es usualmente "tcp" ó "udp".

wait-status — El valor de este campo es "wait" ó "nowait". Generalmente, pero no siempre, servidores de tipo datagrama requieren "wait", y servidores de tipo stream permiten "nowait".

uid — El nombre de usuario bajo el que corre el servidor. Este es cualquier nombre de usuario

válido, pero es normalmente root.

server	Este es el nombre de ruta completo del programa servidor iniciado por inetd.
arguments	Este es cualquier argumento de línea del comando que debe ser pasado al programa servidor cuando es invocado. Esta lista siempre inicia con argv[0] (el nombre del servidor).

No existen muchas situaciones en las cuales necesite modificar el archivo inetd.conf. No todas son requeridas en cada sistema y por razones de seguridad puede desear desactivar servicios no esenciales en algunas computadoras. Para desactivar un servicio, coloque # al inicio de su entrada (lo cual convierte la entrada en un comentario) y pase una señal hang-up (HUP) al servidor inetd. Cuando inetd recibe la señal, lee el archivo de configuración y la nueva configuración toma efecto inmediatamente. Puede también necesitar añadir un servicio que ha sido previamente deshabilitado.

En algunas situaciones, puede necesitar modificar el nombre de ruta de un servidor o de un argumento pasado a un servidor cuando es invocado.

CATITULO 3 Configuración de las Interfaces Ethernet con el Programa ifconfig

El comando ifconfig establece, o checa, los valores de configuración de las interfaces de red. Sin importar el proveedor o la versión de UNIX, el comando ifconfig es utilizado para colocar la dirección IP, la máscara de subred y la dirección broadcast para cada interface. Su función básica es la de asignar la dirección IP.

El administrador de la red debe proveer los valores para la dirección, la máscara de subred, y la dirección broadcast. El nombre de la interface, primer argumento en cada línea de comando ifconfig, debe ser obtenida mediante una herramienta de software, netstat, nos dice qué interfaces están disponibles en el sistema.

3.1. Determinación de la Interface con netstat

Para verificar el estado de todas las interfaces disponibles en la red, tecleamos:

% netstat -ain

La opción -i le dice a netstat que despliegue el estado de las interfaces configuradas de la red. La opción -a modifica el comando para incluir todas las interfaces de la red, no sólo las ya configuradas; la opción -n le dice a netstat que despliegue su salida en forma numérica. El comando netstat -ain despliega los siguientes campos:

Name	El campo del nombre de la interface muestra el nombre actual asignado a la interface. Este es el nombre que se le da a ifconfig para identificar la interface. Un asterisco (*) en este campo indica que la interface no está habilitada, es decir, la interface no está "up".
Mtu	La Unidad de Transmisión Máxima muestra el arreglo (paquete) más largo que puede ser transmitido por esta interface sin ser fragmentado. El MTU es desplegado en bytes.
Net/Dest	El campo Red/Destino muestra la red o el host destino al cual la interface provee acceso. En la mayoría de los casos, este campo contiene una dirección de red. La dirección de red es derivada de la dirección IP de la interface y de la máscara de subred. Este campo contiene una dirección de host sólo si la interface fue configurada para un enlace punto a punto (host específico). Un enlace punto a punto es una conexión directa entre dos computadoras. Se puede crear un punto a punto con el comando ifconfig. Para hacer esto, se coloca la dirección destino directamente después de la dirección de la interface local en la línea de comando ifconfig. La dirección destino es la dirección del host remoto al final del otro enlace punto a punto.
Address	El campo de la dirección IP muestra la dirección Internet asignada a esta interface.
Ipkts	El campo de Paquetes de Entrada muestra cuántos paquetes han sido recibidos por esta interface.
Ierrs	El campo de Errores de Entrada muestra cuántos paquetes dañados han sido recibidos por esta interface.
Opkts	El campo de Paquetes de Salida muestra cuántos paquetes han sido enviados por esta interface.
Oerrs	El campo de Errores de Salida muestra cuántos paquetes han provocado una condición de error.
Collis	El campo de Colisiones muestra cuántas colisiones Ethernet fueron detectadas por esta interface. Las colisiones Ethernet son una condición normal causada por la contención de tráfico Ethernet. Este campo no es aplicable a interfaces no Ethernet.
Queue	EL campo de Fila de Paquetes muestra cuántos paquetes están en la cola, esperando su transmisión a través de esta interface. Normalmente este campo es cero.

El campo Net/Dest cambia por Network y no aparece el campo Queue, al desplegarse el comando netstat -ian en las versiones AIX, IRIX, SCO UNIX, y HP-UX. Mientras que en las versiones SunOS y Solaris aparecen los campos tal como están mostrados arriba. En Linux el comando netstat -ain despliega los siguientes campos:

Iface	Es el nombre de la interfaz de red.
MTU	Es la cantidad de bytes más grande que puede enviarse en una transmisión por esta interfaz.
RX-OK	Es la cantidad de paquetes recibidos sin error.
RX-ERR	Es la cantidad de paquetes recibidos con error.
RX-DRP	Es la cantidad de paquetes caídos.
RX-OVR	Es la cantidad de errores de desbordamiento de paquete.
TX-OK	Es la cantidad de paquetes transmitidos sin error.
TX-ERR	Es la cantidad de paquetes transmitidos con error.
TX-DRP	Es la cantidad de paquetes caídos durante la transmisión.
TX-OVR	Es la cantidad de paquetes caidos por errores de desbordamiento
Flags	Las siguientes banderas pueden aparecer en este campo
	A La interfaz recibe paquetes de direcciones multicast.
	B La interfaz recibe paquetes emitidos.
	D Está activada en la actualidad la característica de depuración de la interfaz.
	L Es la interfaz de loopback.
	M La interfaz está en modo libre.
	N La interfaz no procesa las continuaciones de paquetes.
	O El protocolo de resolución de direcciones está desactivado para esta interfaz.
	P Esta interfaz se está usando como una conexión punto a punto.
	R La interfaz se está ejecutando.
	U La interfaz se ha activado.

3.2. Verificación de la Interface con ifconfig

Muchos sistemas, utilizan un guión de instalación para instalar UNIX. Este guión solicita la dirección del host, la cual utiliza para configurar la interface de red. Sin embargo, esta configuración puede no ser lo que necesitamos exactamente. Se puede verificar la configuración de la interface con ifconfig. Para desplegar los valores actuales asignados a la interface, tecleamos ifconfig con el nombre de la interface y ningún otro argumento.

ifconfig nombre_de_la_interface

Cuando se utiliza para verificar el estatus de una interface, el comando ifconfig despliega dos líneas de salida. La primera línea muestra el nombre de la interface, y las banderas que definen las características de la interface que tiene un valor numérico, el cual corresponde a:

UP	La interface está habilitada para su uso.
BROADCAST	La interface soporta broadcast; significa que está conectada a una red que soporta broadcast, tal como una Ethernet.
NOTRAILERS	Esta interface no soporta encapsulación de remolque. Esta es una característica específica de Ethernet, la cual veremos más adelante.
RUNNING	Esta interface es operacional (está funcionando).

La segunda línea en la salida de ifconfig muestra información directamente relacionada con TCP/IP. La palabra "inet" está seguida por la dirección Internet asignada a esa interface. Después, sigue la palabra

"netmask", seguida por la máscara de subred escrita en hexadecimal. Finalmente, la palabra "broadcast" y la dirección broadcast.

3.3. Asignando una Mascara de Subred

Para funcionar correctamente, cada interface en una red física específica debe tener la misma máscara de subred. Para asignar una máscara de subred, se escribe el valor de la máscara después de la palabra "netmask" en la línea de comando de ifconfig. La máscara de subred suele escribirse en la forma "decimal punteada" utilizada por las direcciones IP. Poner el valor de la máscara directamente en la línea de comando de ifconfig es la forma más común, y generalmente la más simple, para asignar la máscara de subred a una interface.

ifconfig nombre_de_la_interface netmask valor_de_la_máscara_de_subred

Poner el valor de la máscara directamente en la línea de comando de ifconfig es la forma más común, y generalmente la más simple, para asignar la máscara de subred a una interface. Pero es también posible para ifconfig tomar el valor de la máscara de subred de un archivo en lugar de la línea de comando, para esto debemos colocar el nombre de la red separada por un espacio en blanco de el valor de la máscara de subred correcta para nuestro sistema en el archivo /etc/networks.

nombre_de_la_red valor_de_la_máscara_de_subred

Una vez que esta entrada ha sido añadida, se puede utilizar el nombre de la red en la línea de comando ifconfig, en lugar de la máscara actual. Conceptualmente, esto es similar a utilizar un nombre de host en lugar de una dirección IP.

ifconfig nombre_de_la_interface netmask nombre_de_la_red

En sistemas SunOS, Solaris y AIX se puede utilizar también */etc/netmasks* para colocar la máscara de subred. El archivo /etc/netmasks es una tabla con entradas de una línea, cada una conteniendo una dirección de red separada por un espacio en blanco de una máscara de subred.

dirección_de_red máscara_de_subred

El signo más después de la palabra "netmask" provoca que ifconfig tome el valor de la máscara de /etc/netmasks. Ifconfig busca en el archivo la dirección de red que coincida con la dirección de red de la interface que está siendo configurada. Entonces, éste extrae la máscara de subred asociada con esa dirección y la aplica a la interface.

ifconfig nombre_de_la_interface netmask +

3.4. Colocación de la Dirección Broadcast

En el momento en que la dirección Broadcast es definida, ifconfig deberá ser capaz de computarla automáticamente, y podremos utilizarla por default en casi cualquier situación. Desafortunadamente, este no es el caso. TCP/IP fue incluido en BSD 4.2, antes que se definiera claramente el formato de la dirección broadcast como una dirección con todos los bits de host en uno, y fuera adoptado como estándar. BSD 4.2 utilizaba una dirección broadcast con todos los bits de host en cero y no permitía que la dirección broadcast fuera modificada durante la configuración. Debido a esta historia, algunas versiones actuales de UNIX ponen por default la dirección broadcast en "estilo-cero" para compatibilidad con los sistemas antiguos, mientras que otras versiones ponen por default a la dirección broadcast "estándar estilo uno".

Para evitar confusiones definimos la dirección broadcast para toda la red y asegúramos que cada dispositivo de la red lo defina explícitamente durante la configuración. Colocamos la dirección broadcast en el comando ifconfig utilizando la palabra broadcast seguida por la dirección broadcast correcta. Nótese que la dirección broadcast está relacionada con la subred local.

ifconfig nombre_de_la_interface broadcast dirección_broadcast

3.5. Asignación de la Dirección de la Interface de Red

Si sólo se desea asignar la dirección IP a una interface de red, tecleamos ifconfig con sólo el nombre de la interface y la dirección IP. Utilizando la dirección IP que el administrador de la red haya asignado a la interface dada.

ifconfig nombre_de_la_interface dirección_IP

Podemos utilizar el nombre del host en lugar de la dirección en el comando ifconfig.

ifconfig nombre_de_la_interface nombre_del_host

Una dirección IP es preferible, porque si un nombre de host es usado, ifconfig debe resolver el nombre del host antes que la dirección que va aser asignada a la interface. El sistema debe ser capaz de encontrar el nombre del host en /etc/host porque ifconfig se ejecuta antes de que sea ejecutado DNS. Si se decide a utilizar el nombre del host, debemos colocar el nombre del host y la dirección IP en el archivo /etc/host.

nombre_del_host dirección_IP

En la mayoria de los sistemas, la interface loopback es parte de la configuración por default, así que generalmente no necesita ser configurada. Si fuera necesario configurar lo0 en el sistema, utilizamos el siguiente comando:

 # ifconfig lo0 127.0.0.1

3.6. Colocación de ifconfig en los Archivos de Inicio

El comando ifconfig es ejecutado normalmente en el arranque por el archivo de inicio. En sistemas UNIX BSD como SunOS, los comandos ifconfig se localizan generalmente en el archivo /etc/rc.boot ó en /etc/rc.local. El Sistema V de UNIX presenta un grupo más complejo de archivos de inicio, pero los estatutos de ifconfig se localizan generalmente en un archivo con un nombre como el de /etc/tcp en UNIX SCO; /etc/init.d/rootusr ó en /etc/init.d/inetsvc en Solaris; /etc/init.d/network en IRIX; /etc/rc.d/rc.inet1 en Linux; /etc/rc.net en AIX; y el de /etc/netlinkrc en HP-UX. Como el acceso a la red es importante para algunos de los procesos que ejecutan los archivos de inicio, los estatutos de ifconfig se ejecutan cerca del comienzo del proceso de inicio. La forma más simple de configurar una interface de red que cumpla con los requerimientos necesarios es la de editar los archivos de inicio e insertar los estatutos de ifconfig correctos. No borramos ninguna linea ifconfig, tal como la linea para lo0, a menos que se reemplace por otra.

Algunos sistemas toman ventaja de el hecho de que la dirección IP, la máscara de subred, y la dirección broadcast pueden ser colocadas indirectamente para así reducir la magnitud que los archivos de inicio necesitan para ser utilizados Reduciendo su utilización disminuye la posibilidad de que un sistema quede suspendido mientras se levanta porque el archivo de inicio fue editado erróneamente, y esto hace posible preconfigurar estos archivos para todos los sistemas de la red. Los archivos /etc/hosts, /etc/networks y

/etc/netmasks que proveen entradas al comando ifconfig, producen mapas NIS que pueden ser administrados centralmente en sitios que utilizan NIS.

Una desventaja de colocar los valores de ifconfig indirectamente es que puede hacer el soporte técnico más engorroso. Si todos los valores se colocan en los archivos de inicio, sólo se necesitarán verificar en éstos. Cuando la información de configuración de la red es administrada indirectamente, se necesitará verificar el archivo de inicio, el archivo hosts, el archivo networks y el archivo netmasks para encontrar los problemas. Un error en alguno de estos archivos puede provocar una configuración incorrecta. Para hacer la depuración más fácil, muchos administradores de sistemas prefieren colocar los valores de configuración directamente en la línea de comando de ifconfig.

3.7. Otras Opciones del Comando

Hemos utilizado ifconfig para colocar la dirección de la interface, la máscara de subred y la dirección broadcast. Ciertamente éstas son las funciones más importantes de ifconfig, pero también tiene otras funciones. También se utiliza para habilitar ó inhabilitar la " encapsulación de remolque ", el ARP y la interface misma. Ifconfig provee también de un "ruteo métrico" utilizado por el Routing Information Protocol (RIP).

CAPITULO 4 Configuración del Ruteo

Una distinción debe ser hecha entre ruteo y protocolos de ruteo. Todos los sistemas rutean datos, pero no todos los sistemas corren protocolos de ruteo. Ruteo es el acto de transportar datagramas basados en la información contenida en la tabla de ruteo. Protocolos de ruteo son programas que intercambian la información utilizada para construir las tablas de ruteo.

La configuración de ruteo de una red específica no siempre requiere un protocolo de ruteo. En situaciones donde la información de ruteo permanece sin cambio, por ejemplo, cuando existe únicamente una posible ruta, el administrador del sistema usualmente construye la tabla de ruteo manualmente. Algunas redes no tienen acceso a otras redes TCP/IP, y por lo tanto no requieren una acción especial del administrador del sistema para construir la tabla de ruteo (ya sea manualmente o con protocolos de ruteo). Las tres configuraciones de ruteo más comunes son:

ruteo mínimo — Una red completamente aislada de todas las demás redes TCP/IP requiere únicamente de un mínimo ruteo. Una tabla de ruteo mínima es construida por ifconfig cuando la interface de red está configurada. Si la red no tiene ningún acceso directo a otras redes TCP/IP, y si no está utilizando subredes, esta puede ser la única tabla de ruteo que necesite. Las redes aisladas no son tan raras como se podría pensar. En el ambiente UNIX, es común para una red TCP/IP de área local tener acceso al mundo exterior sólo a través de UUCP.

ruteo estático — Una red con un número limitado de gateways a otras redes TCP/IP puede ser configurada con ruteo estático. Una tabla de ruteo estático es construida manualmente por el administrador del sistema utilizando el comando route. Las tablas estáticas de ruteo no se ajustan a los cambios de la red, así que pueden ser utilizadas únicamente donde las rutas no cambian. Pero cuando los destinos remotos sólo pueden ser alcanzados a través de una ruta, un ruteo estático es la mejor opción.

ruteo dinámico — Una red con más de una ruta posible al mismo destino debe utilizar ruteo dinámico. Una tabla de ruteo dinámico es construida de la información intercambiada por los protocolos de ruteo. Los protocolos están diseñados para distribuir la información que ajusta dinámicamente las rutas para reflejar los cambios en las condiciones de la red. Los protocolos de ruteo manejan situaciones de ruteo complejas más rápidamente y con mayor precisión de lo que puede hacerlo el administrador del sistema. Los protocolos de ruteo no sólo están diseñados para cambiar a una ruta de respaldo cuando la ruta primaria es inoperable; también están diseñados para decidir cual es la "mejor" ruta hacia un destino. En cualquier red donde existen múltiples caminos para el mismo destino, un protocolo de ruteo debe ser usado.

Las rutas son construidas con ifconfig, manualmente por el administrador del sistema o dinámicamente por los protocolos de ruteo. Pero no importa como son introducidas las rutas, todas terminan en la tabla de ruteo.

4.1. Tabla Mínima de Ruteo

La tabla mínima de ruteo es el contenido de la tabla de ruteo construida por ifconfig cuando las interfaces de red para el host fueron configuradas. Observando el campo de Flags para cada entrada del comando netstat -nr. Todas las entradas tienen la bandera U (up) encendida, indicando que están listas para ser utilizadas, pero ninguna entrada tiene la bandera G (gateway) encendida. La bandera G indica que un gateway remoto es utilizado. La bandera G no está encendida porque ambas rutas son rutas directas a través de interfaces locales, no a través de gateways externos.

La ruta loopback tiene encendida también la bandera H (host). Esto indica que únicamente un host puede ser alcanzado a través de esta ruta. El significado de esta bandera es más claro cuando se observa el campo destino de la entrada loopback. Este muestra que el destino es una dirección de host, no una dirección de red. La dirección de red para loopback es 127.0.0.0. La dirección destino mostrada (127.0.0.1) es la dirección de localhost, como un host individual. Esta ruta de host particular está en cada tabla de ruteo.

Aunque cada tabla de host tiene su ruta de host específica, la mayoría de la rutas son rutas a redes. Una razón por la que las rutas de redes son utilizadas es para reducir el tamaño de la tabla de ruteo. Una organización puede tener solamente una red pero cientos de hosts. Internet tiene unas cuantas miles de redes pero cientos de miles de hosts. Una tabla de ruteo para cada host sería inmanejable.

La capacidad limitada de una tabla de ruteo es fácilmente verificada con el comando ping. ping utiliza el Mensaje Eco ICMP para forzar a un host remoto a repetir un paquete de regreso al host local. Si los paquetes pueden viajar hacia el host remoto y de regreso, esto indica que los dos hosts pueden comunicarse exitosamente. Desplegando una línea de salida para cada ICMP ECHO_RESPONSE recibido (ping en sistemas SunOS y Solaris solamente desplegarán el mensaje is alive si la opción -s no fue usada. Las demás implementaciones ping no requieren la opción -s.). Cuando ping es interrumpido, despliega algunas estadísticas totales. Todo esto indicando comunicación exitosa con el host remoto.

ping -s

Cuando ping despliega una línea de salida con el mensaje "no answer from", indica que el host local no sabe como enviar datos a la red en la que se encuentra el host remoto.

4.2. Construcción de una Tabla de Ruteo Estático

La tabla mínima de ruteo sólo funciona para alcanzar hosts en las redes que se encuentran conectadas directamente de forma física. Rutas a través de gateways externos deben ser añadidas a la tabla de ruteo para alcanzar hosts remotos. Una forma de hacer esto es construyendo una tabla de ruteo estática con comandos route.

Utilizamos el comando UNIX route para añadir o eliminar manualmente entradas en la tabla de ruteo. La primera palabra en cada línea con el comando route es add o delete, indicándole a route añadir una nueva ruta o eliminar una existente respectivamente. Aquí no existe una palabra por default; una de esas palabras debe ser utilizada.

El siguiente valor es la dirección destino, la cuál es la dirección a alcanzar a través de esta ruta. La dirección destino puede ser especificada como una dirección IP, un nombre de red del archivo /etc/networks, el nombre de un host del archivo /etc/hosts o la palabra default. Debido a que la mayoría de las rutas son añadidas en el proceso de inicialización, las direcciones numéricas IP son utilizadas mas que los nombres. Esto hace que la configuración de ruteo sea independiente del estado del servidor de nombres. Siempre utilizamos una dirección numérica completa (los cuatro bytes completos). Si introducimos menos de cuatro bytes, route expandirá la dirección, y la dirección expandida puede no ser la que uno deseaba.

Si la palabra default es utilizada para la dirección destino, route crea una ruta por default. La ruta por default es utilizada cuando no existe una ruta específica hacia un destino, y a menudo es la única ruta que se necesita. Si la red tiene únicamente un gateway, utilizamos una ruta por default para direccionar todo el tráfico destinado para redes remotas a través de ese gateway.

Lo siguiente en la línea de comando route es la dirección del gateway. Esta es la dirección IP del gateway externo a través del cual los datos son enviados a la dirección destino. La dirección debe ser la dirección de un gateway en una red conectada directamente. Las rutas TCP/IP especifican el siguiente salto en el camino

hacia un destino remoto. El siguiente salto debe ser directamente accesible al host local; por lo tanto, este debe estar en una red conectada directamente.

El último argumento en la línea de comando es la métrica del ruteo. El argumento métrica no es usado cuando las rutas son eliminadas, pero es requerido en cualquier momento que una ruta es añadida. A pesar de ser requerida, route sólo utiliza la métrica para decidir si esta es una ruta a través de una interface directamente conectada o una ruta a través de un gateway externo. Si la métrica es 0, la ruta es instalada como una ruta a través de una interface local, y la bandera G, la cual vimos en el desplegado de netstat -i, no es encendida. Si el valor de la métrica es mayor que 0, la ruta es instalada con la bandera G encendida, y la dirección del gateway es asumida como la dirección de un gateway externo. El ruteo estático no hace uso de la métrica. El ruteo dinámico es requerido para hacer un uso real de los valores variantes de la métrica.

Para añadir la ruta de una red a la tabla de ruteo del host, por medio de un gateway externo, introducimos:

route add dirección_destino dirección_del_gateway métrica_del_ruteo

El comando route anterior funciona para las versiones SunOS, Solaris, AIX, SCO UNIX, HP-UX, e IRIX. Los argumentos de la línea de comando que usa el comando route en Linux cambian en la dirección del gateway, ya que hay que adicionarle la palabra gw antes de la dirección del gateway especificado, y en la métrica del ruteo, porque esta opción todavía no está implantada.

route add dirección_destino gw dirección_del_gateway

El siguiente comando adiciona una ruta estática a una subred vía un gateway exterior, a la tabla de ruteo de un host que pertenece a la misma red, usando el argumento net para hacer que la dirección especificada se trate como una dirección de red. Esto es utilizado en las versiones SunOS, Solaris, AIX, SCO UNIX, HP-UX, y IRIX:

 # route add net dirección_de_la_subred_destino dirección_del_gateway métrica_del_ruteo

AIX tiene una pequeña diferencia para este comando:

 # route add -net dirección_de_la_subred_destino dirección_del_gateway métrica_del_ruteo

Para decirle a Linux la manera de llegar a otra subred, se necesitan este registro de tabla de ruteo para estar seguro:

 # route add -net dirección_de_la_subred_destino gw dirección_del_gateway

4.2.1. Instalando Rutas Estáticas al Arranque

Veamos, como debemos añadir el ruteo estático para la ruta por default en la tabla de ruteo al momento del arranque, en las diferentes versiones de UNIX utilizadas. Para esto necesitamos modificar diferentes archivos de inicialización y añadir las declaraciones route que se desean a un archivo de arranque.

Para Solaris y SunOS simplemente adicionamos una entrada para el ruteador dentro de la red en el archivo /etc/defaultrouter:

132.248.102.254

Para HP-UX editamos /etc/netlinkrc para añadir las declaraciones siguientes:

#

```
# Initialize network routing.
#
# (STEP 2) (OPTIONAL, FOR NETWORKS WITH GATEWAYS ONLY)
#
# The route(1m) command manipulates the network routing tables.
# The "case $NODENAME" construct below allows each node in a diskless
# cluster to execute node specific route calls if necessary.  Add entries
# to the case construct for specific nodes in the diskless cluster if needed.
# The STATUS checking is for Instant Ignition.
#
# For example.
#
#   case $NODENAME in
#     $ROOTSERVER) /etc/route add 192.0.2 gatenode 1
#                  STATUS=$?
#                  if [ ! $STATUS -eq 0 ]
#                  then
#                    net_init=1
#                  fi
#                  ;;
#                *) /etc/route add default 15.2.104.69 1
#                  STATUS=$?
#                  if [ ! $STATUS -eq 0 ]
#                  then
#                    net_init=1
#                  fi
#                  ;;
#   esac
#
# adds network destination "192.0.2" to the rootserver's routing tables,
# indicating a correspondence between that destination and the gateway
# "gatenode", and specifying the number of hops to the gateway as 1.  For
# all other nodes (* is the wildcard), the default gateway is set to
# 15.2.104.69.
#
# The route command should be invoked once per gateway.
#
# SEE ALSO: route(1m), routing(7)

case $NODENAME in
          *) /etc/route add default 132.248.102.254 1
                  STATUS=$?
                  if [ ! $STATUS -eq 0 ]
                  then
                    net_init=1
                  fi
                 ;;
esac
```

Para Linux, primero editamos el guión /etc/rc.d/rc.inet1 para cambiar el valor de la variable GATEWAY por la dirección de nuestro gateway ó ruteador:

```
# IF YOU HAVE AN ETHERNET CONNECTION, use these lines below to configure the
# eth0 interface. If you're only using loopback or SLIP, don't include the
```

```
# rest of the lines in this file.

# Edit for your setup.
IPADDR="132.248.102.21" # REPLACE with YOUR IP address!
NETMASK="255.255.255.0" # REPLACE with YOUR netmask!
NETWORK="132.248.102.0" # REPLACE with YOUR network address!
BROADCAST="132.248.102.255"     # REPLACE with YOUR broadcast address, if you
                # have one. If not, leave blank and edit below.
GATEWAY="132.248.102.254"       # REPLACE with YOUR gateway address!
```

después, quitamos los comentarios para inicializar la tabla de ruteo IP:

```
# Uncomment these to set up your IP routing table.
/sbin/route add -net ${NETWORK} netmask ${NETMASK}
/sbin/route add default gw ${GATEWAY} metric 1
```

Para IRIX, los comandos para adicionar rutas estáticas deben ser creados en un separado guión llamado /etc/init.d/network.local, editándolo para añadir la declaración route:

```
route add default 132.248.102.254 1
```

haciendo después ligaduras simbólicas en /etc/rc0.d y /etc/rc2.d de este archivo para que sea llamado durante el arranque y a la baja del sistema:

```
ln -s /etc/init.d/network.local /etc/rc0.d/K39network
ln -s /etc/init.d/network.local /etc/rc2.d/S31network
```

Para UNIX SCO, primero editamos /etc/tcp para añadir las declaraciones route:

```
/etc/route add default 132.248.102.254 1
```

después, comentar las líneas que inician el protocolo de ruteo routed:

```
#       if [ -x /etc/routed ]; then
#               routed  ; echo "routed \c"
#       fi
```

Para AIX simplemente editamos /etc/rc.net para quitar el comentario en las declaraciones route, y cambiar la palabra gateway por la dirección IP del ruteador por default:

```
# /usr/sbin/route add 0 gateway                 >>$LOGFILE 2>&1
```

Estos simples pasos son todo lo que se necesita hacer para configurar el ruteo estático. El problema con el ruteo estático no es configurarlo, sino darle mantenimiento si se tiene un ambiente de red sujeto a modificaciones. Protocolos de ruteo son lo suficientemente flexibles para manejar ambientes de ruteo simples y complejos. Es por esto que los procedimientos de arranque frecuentemente corren protocolos de ruteo por default.

Utilizamos únicamente el ruteo estático, por que los hosts de cada subred usan este para llegar a sus vecinos inmediatos, siendo suficiente el comando route a fin de ajustar rutas estáticas en cada host al momento del arranque. La ruta por default, usada para paquetes que no encuentran ninguna otra ruta en la tabla de ruteo, se establece para un gateway que ejecuta el ruteo dinámico y sabe acerca del resto del mundo.

CAPITULO 5 Configuración del Servicio de Nombre de Dominio (DNS)

El DNS proporciona un mecanismo de asignación de nombre de host a dirección IP, que es efectivo y relativamente transparente. El primer paso para usar el DNS es configurar la biblioteca del resolvedor en la computadora particular. Uno debe configurar el resolvedor local si se pretende usar una resolución de nombres DNS, aunque no se piense ejecutar un servidor de nombre de dominio local.

5.1. Estructura del Espacio de Nombre de Dominio

Como el nombre DNS indica, el espacio de nombre de dominio de una red esta dividido en dominios. Los dominios están arreglados en estructura arborecente; esto es una raíz y sobre de eso están los llamados dominios de nivel alto, que están otra vez subdivididos en subdominios. Un dominio esta referido especificando una ruta desde la rama en dirección de la raíz, por ejemplo, CS.PURDUE.EDU. Al final de las ramas están objetos direccionables tales como computadoras host y mailbox. Los nombres de estos objetos son siempre únicos con un dominio.

Los dominios de alto nivel están predefinidos por el NIC, estos son dominios de usuario en U.S.A. Así, esto incluye:

- MIL para la fuerza militar de E.U.
- EDU para universidades y otras organizaciones educacionales.
- GOV para organizaciones de gobierno.
- COM para organizaciones industriales comerciales.
- ARPA para organizaciones especificas Internet.

En adición a estos dominios de alto nivel para organizaciones en USA, hay un dominio de alto nivel para cada país conectado, por ejemplo, DE para Alemania, UK para Gran Bretaña, MX para méxico, etc. En algunos países, el dominio de alto nivel nacional incluye una estructura de subdominio similar a la de el dominio de alto nivel en U.S.A.

5.2. Servidores de Nombre DNS

Los servidores de nombre DNS manejan las llamadas zonas. Una zona empieza en un nodo en el árbol DNS y contiene todas las ramas. Un servidor de nombre puede delegar autoridad sobre una subzona a otro servidor de nombre y así controlar las fuentes de información en un subdominio. DNS consiste de un gran número de zonas anidadas, en que los servidores de nombre operan. Cada uno de estos servidores de nombre reconoce a sus servidores vecinos en las zonas próximas arriba y abajo. Por razones formales, cada zona tiene al menos dos servidores de nombre activos (primario y secundario), y ambos proveen la misma información.

Información acerca de objetos direccionables es almácenado en un registros de recurso, que son manejados por servidores de nombre. Programas de usuario generan peticiones de registro de recurso vía el resolvedor. Los registros de recurso consiste de los siguientes campos:

[name] [ttl] IN type data

name Este es el nombre del dominio objeto al cual el registro de recurso se refiere. Puede ser un host individual o un dominio entero. La variable dada para name es relativa al dominio actual a menos que termine con un punto. Si el campo name es un espacio en blanco, el registro se aplica al dominio objeto que fue nombrado últimamente.

ttl El campo tiempo de vida (time-to-live) define la duración del tiempo, en segundos, que la información en este registro de recurso debe ser mantenido en el cache. Usualmente este campo se deja en blanco y el ttl default, puesto para toda la zona en el tipo de registro SOA, es usado.

IN Identifica el registro como un registro de recurso de Internet DNS. Hay otras clases de registros, pero no son usados por DNS.

type Identifica que tipo de registro de recurso es. La tabla lista todos los tipos de registro bajo el encabezado " Tipo ". Se debe especificar uno de estos valores en el campo type.

data La información especifica para este tipo de registro de recurso. El formato del campo de datos depende del contenido del campo de tipo. Este valor se requiere.

Tipos de registro de recurso más utilizados

Tipo	Descripción
A	Es un registro de dirección. Asocia un nombre de un host con una dirección. El campo de datos guarda la dirección en un formato decimal con puntos. Nada más puede haber un solo registro A para cualquier host dado y se considera información confiable. Otra asignación de nombre u otra dirección para este host debe darse con el tipo CNAME.
CNAME	Este campo asocia un alias para un host con su nombre canónico, que es el nombre especificado en el registro A para este host.
HINFO	Proporciona información acerca de un host. El campo de dato guarda información del hardware usado y el sistema operativo que esta corriendo para un host particular.
MX	Establece un registro que intercambia correo. El campo de datos guarda un valor de preferencia entero, seguido por un nombre de host. Los registros MX le ordenan al transporte de correo que envíe el correo a otro sistema que sepa la manera de entregarlo a su destino final.
NS	Apunta a un servidor de nombre de otra zona. El campo de dato del registro de recurso NS contiene el nombre DNS del servidor de nombre. Uno necesita especificar también un registro A para que corresponda con el nombre del host que tiene la dirección del servidor de nombre.
PTR	Esto asigna direcciones a nombres, como en el dominio in-addr.arpa. El nombre de host debe ser el nombre de host canónico.
SOA	El tipo de registro Inicio de autoridad o (SOA) le dice al servidor de nombre que todos los registros de recurso que siguen son confiables para este dominio. ( ) rodea el campo de dato, que es casi siempre un campo de varias líneas. El campo de dato del registro SOA contiene estos datos: origin Es el nombre canónico del servidor de nombre primario para este dominio. Por lo

		general, se da como un nombre de dominio absoluto y termina con un punto (.), por lo que no lo modifica named.
	contact	Es el contacto de correo electrónico de la persona responsable de mantener este dominio. Como el carácter @ tiene un significado especial en los registros de recurso, lo reemplaza un carácter de punto (.).
	serial	Es el número de versión del archivo de información de la zona, que se da como un entero. Lo usan los servidores de nombre secundarios, para determinar cuándo ha cambiado el archivo de información de zona. Se incrementa este número en 1, cada vez que se modifique el archivo de información.
	refresh	Es la cantidad de tiempo en segundos que debe esperar un servidor secundario, antes de intentar revisar el registro SOA del servidor de nombre primario. El registro SOA no cambia con mucha frecuencia, por lo que, en general, este valor puede estar listo en un día, aproximadamente.
	retry	Es el tiempo en segundos que un servidor secundario espera para volver a intentar una petición al servidor primario, si el servidor primario no está disponible. Típicamente, es del orden de unos cuantos minutos.
	expire	Es el tiempo en segundos que el servidor secundario debe esperar, antes de desechar la información de zona, si no pudo hacer contacto con el servidor primario. Este número es por lo regular muy grande, de 30 días más o menos.
	minimum	Es el valor ttl por omisión para los registros de recurso que no lo especifican. Si la red no cambia muy seguido, este número puede establecerse en un valor bastante grande, como un par de semanas. Siempre es posible hacerlo a un lado, al especificar un valor ttl en los registros de recurso.
WKS	Lista los servicios de red soportados por el host especificado. El campo de datos guarda la dirección del host en un formato decimal con puntos, seguido por cualquiera de los dos protocolos de nivel de transporte utilizados por el servicio de comunicación TCP o UDP, y la lista de servicios provistos por este host como ftp, telnet, smtp, domain, y así sucesivamente. Cada host solo puede tener no más de dos registros WKS; un registro para TCP y otro para UDP.	

La petición y respuestas a preguntas consisten de cuatro campos de longitud variable:

- Un campo de petición usado para especificar la información requerida.
- Un campo de respuesta conteniendo la información requerida.
- Si el servidor de nombre no es capaz de abastecer la información, el campo AUTHORITY contiene los nombres de servidores de nombre autorizados que tienen la información.
- Un campo adicional para mayor información (opcional) para el iniciador de la petición, por ejemplo, la dirección de el servidor de nombre dado en el campo AUTHORITY.

5.3. Resolvedor

El resolvedor toma las peticiones del servidor de nombre del programa de aplicación y también en el del usuario. El resolvedor es requerido (hasta posiblemente) para almacenar la información localmente obtenida (caching), para que esta información pueda ser usada en peticiones similares sin la necesidad de mayor comunicación con el servidor de nombre. El periodo de almacenaje depende del periodo de validez especificado en el registro de recurso, que determina el largo de tiempo que el resolvedor debe llevar temporalmente la información en ausencia de una renovada petición para esto.

Un resolver debe ser capaz de llamar a peticiones interactivas y así adelantar peticiones que no puedan ser respondidas exitosamente por un servidor de nombre, usando la información en el campo AUTHORITY, para conectarse a otros servidores de nombre. Puede también preguntar al servidor de nombre para ejecutar esta petición interactiva, en cuyo caso el servidor de nombre es responsable del subsecuente avance (esto incrementa la carga del servidor de nombres).

El resolver UNIX disponible es una implementación muy limitada, que no puede ejecutar caching local o peticiones interactivas. Sin embargo, ambas funciones son implementadas en el servidor BIND, y más aún, el resolvedor está configurado de tal forma que las peticiones fuera del dominio pasan al servidor de nombres. El resolvedor esta diseñado para un claro reemplazo de rutinas existentes para buscar archivos en /etc (como /etc/hosts); así de este modo, como una regla, un programa debe ser conectado a una nueva libreria de subrutina y no necesita ser alterado.

Para peticiones, el resolvedor puede usar UDP o TCP, donde UDP, como protocolo estándar, tiene su propia protección de errores (repetición de petición). Peticiones interactivas pueden bajo ciertas circunstancias tomar largo tiempo, así que debe tenerse cuidado para asegurarse que las peticiones no son enviadas muy pronto. Si esto no es posible, TCP debe usarse. Estas facilidades en el resolver UNIX son configurables por el usuario.

Si el NIS de NFS y DNS son usados al mismo tiempo, puede haber una sobrecarga en la información distribuida por los dos servicios (por ejemplo en direcciones de sistema). Algunos sistemas contienen entradas en el archivo de configuración del resolvedor que señala al resolvedor los mapas de NIS.

Existen dos formas de soportar la configuración del resolvedor. Puede usar la configuración por default, o crear una configuración propia usando el archivo resolv.conf. El resolvedor no es un separado y ni distinto proceso; este es una biblioteca de rutinas llamada por el proceso de red. Si el archivo resolv.conf existe, este es leído cada vez que un uso del proceso de resolución comienza. A causa de esto, el archivo a menudo no se crea a menos que sea requerido, y no es requerido por sistemas que corren named. Cualquier sistema que corre named, el lado servidor de DNS, probablemente puede usar la configuración default de resolver.

5.3.1. Configuración del Resolvedor por Default

La configuración por default usa el host local como el servidor de nombre por default. Manejando el nombre de dominio por default desde la variable regresada por el comando hostname. Esto lo hace quitando la parte de la variable antes del primer punto, y usando lo restante de la variable como el nombre de dominio.

hostname es un comando de UNIX usado para checar o poner el nombre del host para el host local. Solo el superusuario puede poner el nombre del host, como en este ejemplo:

 # hostname fescunam.cuautitlan1.unam.mx

Sin embargo, cualquier usuario puede checar el nombre host tecleando el comando hostname sin argumentos:

fescunam% hostname
fescunam.cuautitlan1.unam.mx

Este ejemplo muestra que el comando hostname en fescunam regresa la variable fescunam.cuautitlan1.unam.mx. Si no se encuentra el archivo resolv.conf, el resolvedor toma el primer componente de este como el nombre del host y usa el restante como el nombre de dominio por default. Si hostname solo regresa el nombre del host, un archivo resolv.conf tendrá una entrada valida con el nombre del dominio por default requerida para este.

Algunos sistemas que usan el Network Information System (NIS) que es diferente. Estos usan el comando hostname para poner y checar el nombre del host, como en algunos sistemas UNIX. Sin embargo, ellos también utilizan un comando domainname que pone el nombre del dominio NIS. Estos sistemas usan el valor devuelto por el comando domainname como el dominio por default DNS cuando el archivo resolv.conf no es encontrado. Debido a esto, Sun recomienda que los dominios NIS y DNS usen el mismo nombre. NIS usa este nombre de dominio para crear un directorio dentro de /var/yp donde los mapas NIS son almacenados. Por ejemplo, el dominio DNS para esta subred es cuautitlan1.unam.mx, así que utilizaremos este como nuestro nombre de dominio NIS. NIS entonces creara un directorio /var/yp/cuautitlan1.unam.mx y almacenara los mapas NIS en este.

El comando domainname checa y pone el nombre del dominio NIS. El superusuario de fescunam puede crear cuautitlan1.unam.mx como el nombre de dominio NIS tecleando:

 # domainname cuautitlan1.unam.mx

El nombre de dominio NIS es normalmente configurado al arranque para colocar el comando domainname en uno de los archivos de arranque. En sistemas SunOS, el valor para el nombre de dominio NIS es tomado desde el archivo /etc/defaultdomain. Este archivo es creado por el guión de instalación, y es usado como entrada para el comando domainname en el archivo /etc/rc.local. Como se muestra abajo, este archivo contiene solo el nombre de el dominio NIS.

 # cat /etc/defaultdomain
cuautitlan1.unam.mx

A la vez que NIS es una posible alternativa de DNS para redes no conectadas a Internet, redes conectadas a la Internet necesitan DNS para el servicio de nombre. Sin embargo, NIS provee más que un simple servicio de nombre, de este modo uno puede usar ambos NIS y DNS. Para usar ambos en un sistema SunOS, uno debe hacer un pequeño cambio en /var/yp/Makefile. Al comienzo inmediato de Makefile uno ve:

#
 # Set the following variable to "-b" to have NIS servers use the domain name
 # resolver for hosts not in the current domain.
#B= -b
B=

Removemos la marca de comentario (#) de la declaración B= -b y marcamos la declaración B= con un comentario, como es mostrado a continuación:

#
 # Set the following variable to "-b" to have NIS servers use the domain name
 # resolver for hosts not in the current domain.
B=-b
#B=

5.3.2. Archivo de Configuración del Resolvedor

Si el sistema local no corre named, o si el nombre de dominio no puede ser derivado del nombre del host, entonces se debe usar el archivo resolv.conf. El archivo de configuración tiene algunas ventajas sobre la configuración por default. Define la configuración del sistema claramente, y permite nombrar servidores de nombre de respaldo que se usan si el servidor default no responde. Por lo tanto, a pesar del adelanto adicional, hay algunas situaciones en que el archivo resolv.conf es deseable.

/etc/resolv.conf es un simple archivo para lectura humana. Hay variaciones de sistemas especificos en los comandos usados en el resolv.conf, pero se soportan dos entradas universalmente en los sistemas SunOS, Solaris, HP-UX, Linux, IRIX, SCO UNIX y AIX:

nameserver dirección — Las entradas nameserver identifican, por dirección IP, los servidores que el resolvedor consultara por información de dominio. Los servidores de nombres son consultados en el orden que aparecen en el archivo. Si no se recibe respuesta desde un servidor, el próximo servidor en la lista es buscado hasta que un numero máximo de tres servidores sean buscados. Si no están contenidas entradas nameserver en el archivo resolv.conf o no existe este, todas las peticiones a servidores de nombre son enviadas al host local. Sin embargo, si hay un archivo resolv.conf y contiene entradas nameserver, el host local no es requerido a menos que una entrada apunte al host local. En un host configurado para correr solo el resolvedor, el archivo resolv.conf contiene entradas nameserver, pero ninguna entrada apunta al host local.

domain nombre — La entrada domain define el nombre del dominio por default. El resolvedor anexa el nombre del dominio por default a cualquier nombre host que no contenga un punto. Entonces se usa el nombre host expandido en la petición que envía al servidor de nombre. Por ejemplo, si el nombre del host fesc (el cual no contiene punto) es recibido por el resolvedor, el nombre del dominio por default es anexado a fesc para construir la petición. Si el valor para nombre en la entrada domain es cuautitlan2.unam.mx, el resolvedor en nuestro ejemplo consultara para fesc.cuautitlan2.unam.mx.

La configuración resolv.conf mas común define el nombre del dominio por default, especificamos nuestro servidor de nombre primario de la red como el primer servidor de nombre, y dos servidores de nombre de respaldo. Un ejemplo de esta configuración se muestra abajo:

```
domain cuautitlan2.unam.mx
nameserver 132.248.10.2
nameserver 132.248.1.3
nameserver 192.103.63.100
```

Este ejemplo es basado en la subred 132.248.102.0, donde el nombre del dominio por default es cuautitlan2.unam.mx. Para la subred 132.248.100.0, el nombre del dominio por default es cuautitlan1.unam.mx. Para la subred 132.248.249.0, el nombre del dominio por default es cuautitlan2.unam.mx, igual que el de la subred 132.248.102.0. Otra cosa que proporcionan los servidores de respaldo, es que este sistema pueda usar solo la configuración por default del resolvedor.

Algunos sistemas permiten al administrador del sistema utilizar tanto un archivo de host tradicional como DNS y especificar el orden en como serán consultados (y posiblemente NIS también). Alguna versión de resolv.conf soportan la palabra clave hostresorder como IRIX:

```
hostresorder local bind
```

Esta entrada le dice que consulte el archivo /etc/hosts primero, y después buscar en el sistema de DNS.

En sistemas de Linux, el archivo /etc/host.conf es usado para este propósito:

order hosts, bind

Este archivo le dice que busque en /etc/hosts un desconocido nombre de host y que consulte el sistema de DNS solamente si este no estuviera allí.

El archivo equivalente bajo Solaris es /etc/nsswitch.conf; esta es la entrada equivalente:

hosts: files dns

5.4. Usando nslookup

nslookup es una herramienta de debuger como parte del software de BIND. Le permite a cualquiera consultar directamente un servidor de nombre y recuperar cualquier información conocida por el sistema DNS. Es útil para determinar si el servidor esta corriendo correctamente y si esta propiamente configurado, o para consultar información dada por servidores remotos.

El programa nslookup se usa ya sea para resolver peticiones interactivas o directamente en la línea de comandos. Desde la línea de comando nslookup se usa para consultar direcciones IP de un host. Esto es útil, pero nslookup se usa mas a menudo en modo interactivo.

El poder real de nslookup se ve en el modo interractivo. Para entrar al modo interactivo, tecleamos nslookup en la línea de comandos sin ningún otro argumento. Una sesión interactiva se termina con el comando exit en el prompt (>).

Por default, nslookup consulta para registros A pero se puede usar el comando set type para cambiar la consulta a otro tipo de registro de recurso. En la tabla, se listan los tipos de información que pueden especificarse y el código de tipo adecuado para el comando set type.

Información	Código de tipo
Dirección IP del host (por default)	A
Nombre de alias canónico	CNAME
Información de CPU y sistema operativo	HINFO
Destino de correo	MD
Registros de grupode correo	MG
Registros de intercambiador de correo	MX
Nombre de dominio de correo renombrado	MR
Buzón o información de lista de correo	MINFO
Nombre de servidor para la zona	NS
Cualquier información que se encuentre	ANY

Notese que la consulta que es puesta a otro tipo de registro de recurso, permanece como esta. No revirtiendose la consulta al tipo por default A. Se requiere de otro comando set type para reiniciar el tipo de consulta.

Puede usarse el comando server para controlar el servidor usado para resolver peticiones. Esto es particularmente útil para ir directamente a un servidor autorizado para checar la información.

Usando el comando set domain, ponemos el domino por default a un dominio especifico. nslookup usa este nombre de dominio por default para expandir los nombres hosts en sus consultas, en la misma forma que usa resolver el nombre dominio default definido en *resolv.conf*.

El comando ls solicita una transferencia del archivo de zona y despliega el contenido del archivo de zona que recibe. Si el archivo de zona es un poca mas grande que unas cuantas líneas de largo, redirige la salida a un archivo, y usamos el comando view para examinar el contenido del archivo. (view arregla un archivo y lo despliega usando el comando de UNIX more). La combinación de ls y view son útiles cuando se conecta a un remoto servidor de nombre.

El uso de nslookup nos permite:

- Consultar por cualquier tipo especifico de registro de recurso.
- Consultar directamente a servidores autorizados por un dominio.
- Obtiene el contenido entero de un dominio en un archivo de manera que se pueda ver.

CAPITULO 6 Configuración de el Sistema de Correo Electrónico

Para que el sistema de correo electrónico pueda operar, el administrador de red debe configurar sendmail, que es el demonio responsable de entregar correspondencia electrónica.

El nombre formal para un programa de correo es un Agente de Usuario. Un Agente de Usuario esta diseñado para desempeñar varios quehaceres: mostrar información acerca de mensajes de correo entrantes que están esperando en el mailbox del usuario, salvar mensajes entrantes o de salida en archivos locales, apuntar a un usuario para los recipientes o tema de un mensaje, y proveer de un editor bueno para la entrada de texto del mensaje. Ejemplos de Agentes de Usuario incluyen el tradicional mail y mailx así como los nuevos programas llamados elm y mush.

El estilo de Agente de Usuario que una persona prefiere ha estado siempre visto como una cuestión de gusto personal, y no a una sujeta normalización. Lo importante es que el resultado final son siempre los mismos detalles de correo el envío y la entrega.

Cuando el destinatario del correo electónico es un apodo o alias, no un identificador de recipiente real. El Agente de Usuario busca al destinatario en un archivo de nombres de alias de correo. Si los alias no están normalizados, la mayoría de los productos de correo permiten crear archivos de alias privados de usuarios y también soportan un archivo de alias público que es mantenido por un administrador de correspondencia.

Hay un formato genérico para identificadores de recipientes de correo en Internet. Sin embargo, proveedores de software y proveedores de servicio de correo públicos han expresado una gran cantidad de individualidad al diseñar sus propios formatos de recipientes. Hay gateways de correo que son ocupados para la conversión entre estos formatos. Un programa de transferencia de correspondencia ampliamente disponible nombrado sendmail puede desempeñar conversiones entre montones de diferentes formatos de nombre. smail es una más reciente alternativa. SCO UNIX y otros sistemas también proveen el Multi-channel Memoreandum Distribution Facility (MMDF), que usa un diferente agente.

Generalmente un identificador de recipiente contiene una señal de indicio donde el recipiente está localizado. Una vez que conocemos de donde es el destinatario, tenemos que figurarnos como envía el correo fuera hacia él. Una dirección de este es para transferir el correo a través de una conexión de TCP directa entre el host fuente y el host recipiente. A veces es más conveniente el relevo de la correspondencia a través de uno o más hosts intermediarios.

El Simple.Mail Transfer Protocol (SMTP) es un estándar de Internet recomendado para mover el correo entre computadoras.

Cuando el correo llega, el Agente de Usuario del destinatario necesitará entender elementos del mensaje tal como el identificador del remitente, fecha de envío, o el tema. El Agente tiene que mostrar el contenido del texto correctamente.

El estándar para el formato de los Mensajes de Texto de Internet ARPA ha proveído el patrón duradero utilizado por Internet para mensajes de correo por una década. Sin embargo, este formato solamente soporta mensajes de texto ASCII de 7-bits. Recientemente se definieron extensiones que soportan mensajes multi-parte que pueden contener documentos creados por procesadores de palabras, imágenes, o el audio codificado.

6.1. Modelo para las Operaciones del Correo

El correo es preparado con la ayuda de una aplicación del Agente de Usuario. El Agente de Usuario tipicamente envía las colas de correo a una aplicación separada, llamada un Agente de Transferencia de Mensaje, que es responsables de establecer comunicaciones con hosts remotos y transmitir el correo. El Agente de Usuario y el Agente de Transferencia de Mensaje son términos utilizados en las normas del sistema de mensaje de SMTP.

El correo puede ser enviado directamente entre el transmisor y el destino por Agentes de Transferencia de Mensaje, o retransmitido a través de Agentes de Transferencia de Mensaje intermediarios. Cuando un correo es retransmitido, el mensaje completo es transmitido a un host intermediario, donde es almacenado hasta que pueda ser enviado en un tiempo conveniente. Sistemas de correo que utilizan la retransmisión son llamados sistemas de almacenaje y transmisión.

En el host recipiente, el correo está situado en una cola entrante, y posteriormente es movido a un área de almacenamiento de mailbox del usuario. Cuando un usuario invoca un programa Agente de Usuario, el Agente de Usuario presenta generalmente un sumario de correspondencia entrante que está esperando en el mailbox.

Cuando un host utiliza una conexión directa, este puede asegurarse que el correo ha alcanzado su destino. La retransmisión de correo usa recursos de almacenaje intermedio, y requieren conexiones múltiples. Para una retransmisión de correo, tenemos que diseñar un sistemas de almacenaje y transmisión para el camino de la retransmisión de correo, y si no hiciéramos un trabajo bueno, el correo rondará alrededor de una manera ineficiente.

6.2. Nombres y Dominios de Correo

Recipientes de correo en Internet son identificado por nombres siguiendo el patrón general:

parte-local@ nombre-dominio

El formato de la parte-local está autorizado a variar dependiendo del dominio.

Un caso especial de la forma general es el patrón

userid@host-nombre-dominio

Este es un patrón popular para nombres de Internet. Sin embargo, una organización puede no querer anunciar sus userids y nombres de host al mundo. Utilizando un nombre lógico planeado que sigue el patrón general, *parte-local@ nombre-dominio*, puede mejorar la seguridad de una red. Algunos nombres lógicos también permiten a usuarios adquirir nuevos userids o moverse por diferentes computadoras sin cambiar sus identificadores de correo.

La parte nombre-dominio de un nombre de correo de Internet puede ser un nombre lógico que identifica un dominio de correo, más bien que una computadora. Un Agente de Transferencia de Mensaje busca el nombre-dominio en una base de datos (tipicamente por un Domain Name Server) en orden para descubrir si este es un host intercambiador de correo para que el correo pueda ser retransmitido.

Cuando un correo alcanza un intercambiador de correo, la parte-local puede ser buscada en un archivo alias y convertido a un userid y nombre de host, o no importa que tipo de identificador es usado en la red de destino.

6.3. Configurando el Sistema de Correo

Antes de que el sistema de correo opere, el administrador del sistema tiene que adaptar el archivo /usr/lib/sendmail.cf. Este archivo puede ser muy, muy largo, corriendo cientos de líneas. Las buenas noticias es que hay solamente algunos detalles que necesitan ser configurados antes de que mail sea iniciado y corrido; por supuesto, hay muchos más que uno pueda querer realizar a largo plazo.

Estos son los detalles que necesitara tener o que puede necesitar para configurar:

- Las definiciones especificas del host local y sus características (requeridas).
- El indicar entradas de otros sistemas para el host local de correo de intercambio y los medios para hacer esto (requeridos).
- Los agentes de entrega que serán utilizados por diversos tipos de transporte de correo (opcional).
- El formato y contenidos de los encabezamientos del correo electrónico generados por sendmail (opcional).

El archivo de configuración sendmail es adaptado como una serie de definiciones que lucen bastante crípticas; Estas son las entradas mayores que definen el sistema local y los otros con que intercambiara correo directamente:

Entrada	Qué es lo que define
Dw	Define el nombre de host local.
DD	Define el dominio local.
Cw	Definen otros nombres utilizados por el host local.
Cd	Define otros nombres para el dominio local.
CS	Listan otros host con que intercambiara correspondencia directamente.
DF	El nombre del host de adelanto, que manipula el correo del sistema local cuando no sabe como entregarlo.
CF	Otro nombre del host de adelanto.
DR	El nombre del host de retransmisión, que obtiene correo destinado más allá del dominio local. En nuestro caso, este será el mismo que el host de adelanto.
CV	El nombre del host externo con que remitamos correo (definido solamente en el host de adelanto).
DW	Nombre del host uucp del host local (necesitado solamente si es utilizando uucp para intercambiar correo).
DM	El método para transportar correo afuera de el dominio local (por ejemplo, uucp, tcp, etc.).
OA	Ubicación del archivo de definición aliases (generalmente /etc/aliases).
OQ	Ubicación del directorio de colas del correo (generalmente /var/spool/mqueue).

En general, entradas D definen macros sendmail: las variables guardan un valor único. Las entradas C definen clases: las variables guardan una lista de uno o más valores; las clases y macros igualmente nombrados están relacionado como uno esperaría: las entradas de C y D para el mismo macro guardan un valor y lista única de valores para ese detalle. Por ejemplo, el macro de Dw define el nombre del host local principal, y la clase Cw define una lista de nombres aplicables al host local.

Las entradas F definen una clase a través de los contenidos del archivo especificó como a su argumento. y entradas O definen opciones sendmail.

Supongamos que el nombre del host del sistema local es hamlet y solamente correo local dentro de este sistema único es deseado. En este caso, la primera sección del archivo sendmail.cf se parecerá algo a esto (los cambios locales están en negritas):

```
##################################################################
### local info
##################################################################
CV

# My hostnames
Dwhamlet
Cw $w $?D$w.$D$. sys242

# My domain name
DDexpoa.com
CD $d parabola.com

# Major relay mailer
DM

# Major relay host
DRmailhost
CRmailhost

FS/etc/mailhosts
```

Porque este es un sistema aislado, muchos de los campos referente al adelanto del correo se están dejado espacios. Este archivo establece el host de retransmisión, el lugar donde se enviará el correo que el sistema local no conoce para enviarle, como mailhost. Sin embargo, en el archivo /etc/hosts, ese nombre será un alias para hamlet, y de modo que todo el correo permanecerán dentro del sistema local. El archivo /etc/mailhosts, lista los sistemas con que hamlet directamente intercambiara correo, el cual contendrá el único nombre de host hamlet.

6.3.1. Inicializando Alias de Correo y Listas de Distribución

Para enviar correspondencia a un grupo de usuarios, se tiene que crear primero una lista de distribución conteniendo los nombres de los usuarios de los miembros del grupo. (Nótese que "grupo" es utilizado aquí simplemente para referir a una colección de usuarios. No tiene que y generalmente no corresponderían a un grupo de UNIX.) En sistemas que utiliza la facilidad sendmail, la listas de distribución están almacenado en forma ASCII en un archivo nombrado aliases, que es generalmente localizado en /etc bajo Sistema V y /usr/lib en sistemas BSD. Nuevas listas pueden ser creadas editando este archivo.

Las listas de distribución se especifican utilizando el siguiente formato:

aliasname: username1, username2, username3, . . .

Donde aliasname es el nombre por medio del cual usted referirá a la lista de distribución. El nombre de alias es seguido por dos puntos y la lista de usuarios para incluirlos en lista de distribución, separada por comas. Cualquiera línea comenzando con espacio blanco es interpretada como continuación de una línea, de modo que la lista de nombres de usuarios puede continuar a través de varias líneas. Otro formato popular es colocar cada nombre de usuario en una línea separada para la facilidad de leerlo.

Nótese que los usuarios pueden estar situado en más de una lista. Es también legal para un alias ser un miembro de otro grupo de distribución.

Un nombre de host puede ser añadido al nombre de usuario para especificar el correo en un sistema dado:

usuario:
/usuario@ host

La inicial barra invertida inhibe cualquier alias más alejado del nombre (para evitar saltarse el anterior correo y adelante entre el host siempre).

La facilidad sendmail UNIX no hace acceso al archivo aliases directamente. En lugar de eso, utiliza los archivos binarios aliases.dir y aliases. Estas bases de datos de acceso al azar aceleran el proceso de expansión de alias. Una vez que uno ha editado el archivo aliases, uno tiene que actualizar los archivos binarios utiliza el comando newaliases, newaliases es equivalente a sendmail -bi; esto último puede ser utilizado si newaliases no está proveído:

 # newaliases

Cualquier cambio que uno haga al archiva aliases no tomará efecto hasta que este comando sea ejecutado. Uno puede querer correr newaliases después porque puede tomar algún tiempo para completarse.

CONCLUSIONES

Usar el protocolo TCP/IP en una red permite crecimiento y expansibilidad. Conforme aumente la cantidad de vendedores de computadoras que cambien sus sistemas de protocolos propios por protocolos de comunicación abierta, tambien se incrementará la necesidad para TCP/IP. Los protocolos de TCP/IP y la serie de protocolos relacionados proporcionan los mismos servicios de red que se dieron antes. El soporte de discos virtuales, la impresión en red, el arranque remoto, el ruteo y el soporte a terminales virtuales los ofrece por medio de la serie de protocolos TCP/IP.

Las aplicaciones que están emergiendo ahora y en el futuro requieren que muchos sistemas de computadora diferentes compartan información y programas. Ya sea que estos sistemas estén a lo largo de un pasillo o por todo el mundo, los protocolos TCP/IP pueden ayudar a compartir la información en forma confiable y efectiva.

Existen muchas formas de conectarse con una computadora que esté ejecutando UNIX (a las que se hace referencia como un host). Es factible utilizar terminales o computadoras así como estar ubicados físicamente cerca del host, conectado por medio de un cable o estar al otro lado del planeta conectando con líneas de datos de alta velocidad o líneas telefónicas ordinarias. Esto significa que UNIX puede participar completamente en redes de máquinas que utilizan protocolos de comunicación TCP/IP, tanto si todas las máquinas de la red son sistemas con sistema UNIX o no.

Por lo mensionado anteriormente, se realizo la implementación de una red basada en el protocolo TCP/IP en UNIX en la Facultad de Estudios Superiores Cuautitlan.

Antes de configurar un host para correr TCP/IP, tenemos que tener información para realizar esto. Cuando menos, cada host tiene que tener una única dirección IP y nombre de host. Debemos también decidir en los siguientes detalles antes de configurar un sistema:

Dirección de gateway por default	Si el sistema se comunica con hosts de TCP/IP que no son de su red local, una dirección de gateway por default es necesaria.
Protocolo de ruteo	Si un protocolo de ruteo es utilizado en la red, cada dispositivo necesita conocer qué protocolo es este.
Dirección del servidor de nombres	Para resolver nombres de host en direcciones IP, cada host necesita conocer las direcciones de los servidores de nombre del dominio.
Nombre del dominio	Los host usando el servicio de nombre de dominio tienen que conocer su nombre de dominio correcto.
Máscara de subred	Para comunicarse adecuadamente, cada sistema en una red tiene que utilizar la misma mascara de subred.
Dirección broadcast	Para evitar problemas de emisión, la dirección broadcast de cada computadora en una red tiene que ser la misma.
Versión de UNIX	Como hay diferencias significativas en algunos programas y archivos de configuración entre versiones, es necesario especificar cual versión se esta utilizando.

Si se estuviera configurando una red desde cero, estas decisiones se hacen antes de configurar cualquier sistema. Si se estuviera añadiendo un nuevo sistema a una red existente, nos aseguramos de averiguar las

respuestas con el administrador de red para después poner el sistema en linea. El administrador de red es responsables de crear y comunicar decisiones acerca de configuración de la red global. Un manera simple de esto es para el administrador de red crear una lista corta de información para la administración del sistema. A continuación mostraremos esta lista para cada uno de los sistemas configurados en la red de la Facultad de Estudios Superiores Cuautitlan, utilizados en este trabajo:

Nombre del host	olimpia
Dirección IP	132.248.102.32
Versión de UNIX	SCO ODT v.3.2
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 4
Servicio destinado	Lenguajes de programación, y desarrollo de aplicaciones XWindows, SQL y de red.
Administrador del host	Marco Antonio Cruz Mendoza

Nombre del host	cicc
Dirección IP	132.248.102.30
Versión de UNIX	HP-UX 9
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 4
Servicio destinado	Herramientas de visualización.
Administrador del host	Fco. Chavez Castañeda

Nombre del host	irixcicc
Dirección IP	132.248.102.33
Versión de UNIX	IRIX 6.3
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 4
Servicio destinado	Desarrollo de aplicaciones Web e imagenes en 3D, herramientas de visualización, y edición de video.
Administrador del host	Moises Hernandez Duarte

Nombre del host	fesc
Dirección IP	132.248.102.36
Versión de UNIX	Solaris 2.5
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2
	132.248.1.3
	192.103.63.100
	147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 4
Servicio destinado	Servidor de correo electrónico.
Administrador del host	Moises Hernandez Duarte

Nombre del host	fufimat3
Dirección IP	132.248.102.68
Versión de UNIX	AIX 3.2
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2
	132.248.1.3
	192.103.63.100
	147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 4
Servicio destinado	Lenguajes de programación y servidor de correo electrónico.
Administrador del host	Sergio Barragán

Nombre del host	xel-ha
Dirección IP	132.248.102.21
Versión de UNIX	Linux 1.2.1
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2
	132.248.1.3
	192.103.63.100
	147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 4
Servicio destinado	Herramientas de visualización.
Administrador del host	Moises Hernandez Duarte

Nombre del host	fescunam
Dirección IP	132.248.100.53
Versión de UNIX	SunOS 4.1.2
Máscara de subred	255.255.255.0
Gateway por default	132.248.100.254
Dirección broadcast	132.248.100.255
Nombre del dominio	cuautitlan1.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 1
Servicio destinado	Lenguajes de programación, y servidor de correo electrónico.
Administrador del host	Marco Antonio Cruz Mendoza

Nombre del host	www
Dirección IP	132.248.102.34
Versión de UNIX	IRIX 6.3
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Cómputo Campo 4
Servicio destinado	Desarrollo de aplicaciones Web, herramientas de visualización y servidor Web.
Administrador del host	Moises Hernandez Duarte

Nombre del host	IRIS-CIT
Dirección IP	132.248.100.200
Versión de UNIX	IRIX 5.3
Máscara de subred	255.255.0.0
Gateway por default	132.248.100.254
Dirección broadcast	132.248.255.255
Nombre del dominio	cuautitlan1.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Centro de Investigaciones Teoricas Campo 1
Servicio destinado	Desarrollo de aplicaciones en contaminación ambiental, herramientas de visualización, y lenguajes de programación.
Administrador del host	Vladimir Tchijov

Nombre del host	antara
Dirección IP	132.248.100.11
Versión de UNIX	IRIX 5.3
Máscara de subred	255.255.0.0
Gateway por default	132.248.100.254
Dirección broadcast	132.248.255.255
Nombre del dominio	cuautitlan1.unam.mx
Servidor de nombres	132.248.10.2 132.248.204.1 132.248.1.3
Protocolo de ruteo	no utiliza
Ubicación	Fisicoquimica Campo 1
Servicio destinado	Desarrollo de aplicaciones moleculares, herramientas de visualización y edición de video.
Administrador del host	Alberto Ramirez Murcia

Nombre del host	alkymia
Dirección IP	132.248.100.54
Versión de UNIX	AIX 3.2
Máscara de subred	255.255.255.0
Gateway por default	132.248.100.254
Dirección broadcast	132.248.100.255
Nombre del dominio	cuautitlan1.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Sala de Cómputo Ciencias Quimico Biológicas Campo 1
Servicio destinado	Lenguajes de programación y servidor de correo electrónico
Administrador del host	Alejandro Valdez Santamaria

Nombre del host	nutrius
Dirección IP	132.248.100.55
Versión de UNIX	AIX 3.2
Máscara de subred	255.255.255.0
Gateway por default	132.248.100.254
Dirección broadcast	132.248.100.255
Nombre del dominio	cuautitlan1.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Sala de Cómputo Ciencias Quimico Biológicas Campo 1
Servicio destinado	Lenguajes de programación y servidor de correo electrónico.
Administrador del host	Alejandro Valdez Santamaria

Nombre del host	fufimat2
Dirección IP	132.248.102.69
Versión de UNIX	AIX 3.2
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Sala de Cómputo de IME Campo 4
Servicio destinado	Lenguajes de programación y servidor de correo electrónico.
Administrador del host	Daniel R. Elorreaga Madrigal

Nombre del host	nutrius
Dirección IP	132.248.100.55
Versión de UNIX	AIX 3.2
Máscara de subred	255.255.255.0
Gateway por default	132.248.100.254
Dirección broadcast	132.248.100.255
Nombre del dominio	cuautitlan1.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Sala de Cómputo de Informatica Campo 4
Servicio destinado	Lenguajes de programación y servidor de correo electrónico.
Administrador del host	Federico Vargas Carrillo

Nombre del host	fescunam
Dirección IP	132.248.102.71
Versión de UNIX	Solaris 2.5
Máscara de subred	255.255.255.0
Gateway por default	132.248.102.254
Dirección broadcast	132.248.102.255
Nombre del dominio	cuautitlan2.unam.mx
Servidor de nombres	132.248.10.2 132.248.1.3 192.103.63.100 147.225.1.2
Protocolo de ruteo	no utiliza
Ubicación	Sala de Cómputo de IME Campo 4
Servicio destinado	Lenguajes de programación, y desarrollo de aplicaciones Java, SQL y servidor Web.
Administrador del host	Marco Antonio Cruz Mendoza

Los pasos para configurar y correr TCP/IP en un sistema UNIX son:

- Crear archivos de configuración de red para TCP/IP. Un conjunto de archivos de configuración de directorio /etc controla la conexión de la red TCP/IP en UNIX. Estos archivos le dicen a UNIX cual

es su dirección IP, el nombre de host y el nombre de dominio, además de controlar las interfaces de red.

- Configurar las interfaces Ethernet con el programa ifconfig. El programa ifcogfig da a conocer las interfaces de red como el ciclo de retorno de software y las tarjetas Ethernet, al núcleo de UNIX. Esto debe hacerse antes de que UNIX pueda usarlos. El programa ifcofig también se usa para monitoriar y cambiar el estado de las interfaces de red.

- Especificar rutas y otra información de red mediante el programa de ruta. El ruteo determina la ruta que un paquete toma desde su origen en la red hasta su destino. Esta ruta se determina al buscar la dirección IP de destino en las tablas de ruteo del núcleo y al transmitir al paquete a la máquina indicada, que puede ser o no el destino del paquete. Las máquinas de cada subred usan el ruteo estático para llegar a sus vecinos inmediatos, esto es suficiente usando el comando route, a fin de ajustar rutas estáticas en cada máquina al momento de arranque. La ruta por default, usada para paquetes que no encuentra ninguna otra ruta en la tabla de ruteo, se establece para una máquina gateway que ejecuta el ruteo dinámico y sabe acerca del resto del mundo.

- Configuración del Servicio de Nombre de Dominio (DNS). El DNS proporciona un *mecanismo* de asignación de nombre de anfitrión a dirección IP, que es efectivo y relativamente transparente. El primer paso para usar el DNS es configurar la biblioteca del resolvedor en la computadora particular. Uno debe configurar el resolvedor local si se pretende usar una resolución de nombres DNS. aunque no se piense ejecutar un servidor de nombre de dominio local.

- Configurar el sistema de correo electrónico. Para que el sistema de correo electrónico pueda operar, el administrador de red debe configurar sendmail, que es el demonio responsable de entregar correspondencia electrónica.

- Monitorear y resolver problemas de red. Los problemas de red tienen que ver con lo inesperado. Los problemas requieren frecuentemente un conocimiento que es conceptual más bien que detallado. Los problemas de red son generalmente únicos y a veces difíciles de resolver. El repararlos es una parte importante para mantener un estado de servicio de red confiable.

Configurar una red TCP/IP es una de las tareas más comunes que uno enfrenta cuando administra la red UNIX. En los casos más básicos no es muy compleja; sin embargo, requiere que pensemos un poco sobre el diseño de la red y conozcamos una pequeña cantidad de programas y archivos de configuración.

Las aplicaciones clásicas telnet, ftp, y sendmail, son aún las más populares. Pero la red se está utilizado no simplemente como un enlace de entrega entre dos hosts, si no como una vía para compartir recursos de información. Los servicios de información, los sistemas de almacenamiento de archivos, las bases de datos, y directorios de información son disponibles a través del Internet, disponer de estos servicios es la tarea inmediata a realizar después de implementar una red basada en los protocolos TCP/IP en UNIX., sin embargo es importante señalar que las tendencias de desarrollo actuales en la informática prevén una rápida incorporación de los servicios mas diversos que involucran el desarrollo de estándares a un nivel superior, desde luego sobre una base sólida como lo es la Internet y en particular TCP/IP.

BIBLIOGRAFIA

TCP/IP Network Administration
Craig Hunt
O'Reilly & Associates, Inc.

TCP/IP and ONC/NFS Internetworking in a UNIX Enviroment, Second Edition
Michael Santifaller
Addison-Wesley Publishing Company

TCP/IP Architecture, Protocols, and Implementation
Sidnie Feit
McGraw-Hill, Inc.

Essential System Administration, Second Edition
AEleen Frisch
O'Reilly & Associates, Inc.

TCP/IP Illustrated, Volume 1: The Protocols
W. Richard Stevens
Addison-Wesley Publishing Company

Data Communications, Computer Networks and Open Systems
Fred Halsall
Addison-Wesley Publishing Company

Comunicaciones en Unix
Jean-Marie Rifflet
McGraw-Hill Interamericana

Redes Globales de Información con Internet y TCP/IP.
Principios Básicos,Protocolos y Arquitectura, Tercera Edición
Douglas E. Comer
Prentice Hall Hispanoamericana, S.A.

Linux Edición Especial
Jack Tackett
Prentice Hall Hispanoamericana, S.A.

Redes para Proceso Distribuido Area Local, Arquitecturas, Rendimiento, Banda Ancha
Jesús García Tomás, Santiago Ferrando Girón y Mario Piattini Velthuis
RA-MA Editorial