42

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

"CAMPUS ARAGÓN"

"PANORÁMICA GENERAL DE LOS PRINCIPIOS Y ELEMENTOS BÁSICOS QUE INTERVIENEN EN LA SUPER CARRETERA DE INFORMACIÓN"

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

IVONNE MEJÍA BERZUNZA

ASESOR: ING. ERNESTO PEÑALOZA ROMERO

México . 1996

TESIS CON
FALLA DE ORIGEN

TESIS

COMPLETA

GRACIAS . . .

La culminación de todos nuestros esfuerzos se debe no solo a nuestra capacidad de soñar sino al hecho de contar con la motivación necesaria para realizarlos.

El amor y admiración es uno de los motores que mueve a todo ser humano para apartarse de la conformidad y dejar atrás las circunstancias y todo lo que puede asegurarnos un fracaso. Gracias a mis padres, a Doña Laura, a mis amigos, profesores, y a esa "energía divina" por enseñarme el camino de la evolución.

(IMB)

" Panorámica General de los principios y elementos básicos que intervienen en la Super Carretera de Información"

Objetivos Generales:

- Exponer cuales son los principios básicos que intervienen en el proceso de comunicación y transporte de la Super Carretera de Información. Así mismo, presentar el funcionamiento de los elementos que fungen como parte esencial del Sistema.

- Que la información contenida en este trabajo proporcione las bases para estudios más detallados en la elaboración de aplicaciones para la red.

Capitulado:

CAPITULO IV

CAPITULO V

PRINCIPIOS BÁSICOS DE TELNET, FTP Y E-MAIL EN LA SUPER CARRETERA DE INFORMACIÓN.

Capítulo I
Introducción

Una Carretera es una vía pública que sirve como medio de comunicación, transporte y acceso de un lugar a otro.

La *"Super Carretera de Información"* puede describirse como un extenso grupo de redes de computadoras de todas partes del mundo que se encuentran enlazadas entre sí, y por las cuales transitan millones de paquetes de información. Este medio de comunicación es una inmensa red de cobertura mundial (Mega-red) constituida a su vez por múltiples redes conectadas entre sí, las cuales pueden establecer comunicación e intercambiar mensajes. Este concepto es mejor conocido como *INTERNET*.

Algunas definiciones clásicas de esta gran red son:

- Una gran Red Global compuesta por redes independientes de organizaciones y dependencias de todas partes del mundo basadas en los protocolos TCP/IP
- El conjunto de personas encargadas del desarrollo y mantenimiento de estas redes, así como aquellas que las usan
- El agrupamiento de todos los recursos y medios que se accesan a través de estos arreglos de computadoras

En resumen, podemos decir que Internet es la red de redes más grande del mundo. Está integrada por miles de computadoras independientes pero interconectadas entre sí, las cuales hablan un lenguaje común (Protocolo TCP/IP).

Al referirnos a una red estamos especificándola como una arreglo de computadoras interconectadas entre sí a través de algún medio de comunicación para poder establecer la transmisión de información. De hecho, dos o más computadoras interconectadas, capaces de comunicarse entre sí, forman una red de computadoras. Dos o más redes interconectadas forman una "internetwork" -inter-red (internet).

La mayoría de las redes de computadoras, incluyendo a Internet, utilizan en la transmisión de la información el método de conmutación de paquetes. En efecto, podemos referirnos a Internet como una "Red de conmutación de paquetes". En una red de este tipo, los programas dividen la información o los datos en fragmentos, llamados paquetes, y los transmiten entre las computadoras involucradas en el proceso.

En las redes de conmutación de paquetes la transmisión de los datos puede realizarse en un sólo fragmento, pero en otras ocasiones será necesario hacer la transmisión de esos datos en múltiples fragmentos. Estos paquetes que forman parte de un mensaje pueden seguir diferentes caminos o rutas para alcanzar el mismo destino.

En sí, una red de conmutación de paquetes tendrá múltiples rutas entre varias computadoras y los datos pueden viajar en ambas direcciones. De esta forma, cada paquete debe contener su dirección de destino (muchos paquetes también acarrean la dirección de su computadora de origen) para asegurar su transmisión y evitar su pérdida.

Como hemos visto, Internet es una inter-red o sistema de redes interconectadas, y este sistema conecta redes sin importar su topología (estrella, bus, anillo) o tecnología (Ethernet, ARCnet, IBM Token Ring, ATM, etc.). Estas inter-redes comúnmente utilizan dispositivos especiales como los repetidores, puentes, enrutadores y gateways para realizar la conexión de las redes independientes.

Para poder habilitar la comunicación entre redes estructuradas de computadoras diferentes, las redes que constituyen a Internet están basadas en un conjunto de protocolos estándares (TCP/IP), creados como un acuerdo común (un estándar) para establecer las reglas de comunicación entre las computadoras. De esta forma, no interesa que tipo de red o de computadora se quiera enlazar a Internet, cualquiera que contenga la serie de protocolos TCP/IP podrá incorporarse a ella. Dicho de otra manera, no existe una marca o topología exclusiva para poder hacerlo.

Universidades y colegios de todos los niveles, departamentos de gobierno, grandes corporaciones empresariales, comercios de servicios en-línea, y hasta partidos políticos, están conectados a la red y diariamente hacen uso de ella.

Debido a la gran variedad de temas y recursos que contiene esta red, el acceso de datos comprende desde disciplinas artísticas, educativas, culturales, científicas y hasta lo relativo al entretenimiento; el usuario que tenga una buena conexión a este medio, podrá navegar (accesar ó consultar) en una gran variedad de información.

Toda esta gente tiene la ventaja de poder obtener todo tipo de información que requiera, y hacer uso de servicios que van desde correo electrónico, transferencia de archivos, conexión remota con máquinas de otras redes, manejar sus negocios ,así como publicar cualquier tipo de propaganda o publicidad que sea permitido, etc.

Estadísticas actuales[1] reflejan que hay una cifra cercana a los 3 millones de servidores (computadoras conectadas a Internet) que forman parte de la red, y que equivale a mas de 30 millones de personas que hoy en día hacen uso de este extenso sistema de carreteras de información. Estas cifras se incrementan en un promedio de 1 millón de nuevos usuarios al mes, lo que significa que si ahora la red es extensa, el próximo año será masiva, y visiblemente gigantesca en los años subsecuentes. Tarde o temprano todos nos veremos involucrados en ella, solo es cuestión de tiempo.

Ahora que ya sabemos lo que es la Super Carretera de Información (Internet), veamos como surgió. En este trabajo emplearemos ejemplos tomados de RedUNAM.

1.1.- RESEÑA HISTÓRICA

Internet nació hace aproximadamente 20 años, surgió como resultado de la evolución de un proyecto implantado en Estados Unidos en 1969. Un departamento de defensa conocido como "The Defense Advanced Research Projects Agency" (DARPA), se percato de la necesidad de crear un sistema que facilitara el intercambio de información militar entre científicos e investigadores localizados en diferentes lugares geográficos. Para ello instalaron una red de 4 computadoras (3 en California y 1 en Utah) que a su vez conectaron a otras redes de radio y satélite y la nombraron DARPANET. El sistema siguió en marcha, pero el nombre fue cambiado a **ARPANET** que para 1972 ya contaba con 37 computadoras. Al mismo tiempo el empleo de la red iba cambiando, ya no solo se manejaba información militar que era sumamente importante e igualmente aburrida, los empleados de ARPANET comenzaron a enviar mensajes entre ellos por correo electrónico, algunas veces eran serios y otras muy triviales. Quizá nunca imaginaron lo que estos trabajadores habían comenzado.

---

[1] *Estadísticas obtenidas de páginas de Internet WWW. El Worl Wide Web (WWW) es el medio donde podemos localizar y accesar información de Internet a través de páginas en forma gráfica.*

En el modelo ARPANET la comunicación se efectuaba entre una computadora fuente (transmisora) y otra que era el destino. Para poder enviar un mensaje por la red, la computadora solo colocaba los datos en paquetes denominados "paquetes de protocolo internet" y los direccionaba correctamente a la computadora destino. La filosofía que manejaba era que cada computadora de la red pudiera comunicarse en pareja con cualquier otra.

ARPANET continuo trabajando y para 1983 había crecido en gran medida que fue necesario mover todos los componentes referentes al campo militar a una red especifica llamada MILNET. Al mismo tiempo las redes Ethernet de área local (LAN) empezaban a evolucionar y se lanzaron al mercado las estaciones de trabajo . La mayoría de esas máquinas incluían el sistema operativo Unix de Berkeley y software para red con el protocolo IP (Internet Protocol) usado por ARPANET.

A pesar de los esfuerzos de la "Organización para la Estandarización Internacional (ISO)" por diseñar el modelo estándar de comunicación entre computadoras de una red, los usuarios y desarrolladores de Internet de países como Estados Unidos, Reino Unido y Escandinavia, no podían seguir esperando y en respuesta a la demanda del mercado, iniciaron la adición de software del protocolo de ARPANET (IP) en cada tipo de computadora. Este método hacía factible que máquinas de diferentes y diversos fabricantes , pudieran trabajar en una misma red y establecer comunicación entre ellas. Este software las hacia compatibles.

Rápidamente muchas otras organizaciones implantaron sus propias redes utilizando los mismos protocolos de comunicación de ARPANET (IP) . Era obvio que si estas redes podían comunicarse entre ellas, entonces los usuarios de una red podrían comunicarse con aquellos pertenecientes a otra. Una de estas nuevas organizaciones era "The National Science Foundation (NSF)", que en 1984 estableció la NSFNET. Esta red interconectaba 5 grandes centros de cómputo que ahora permitían a instituciones educativas el acceso a estos recursos.

NSF instalo su red con la tecnología y protocolos de ARPANET y conecto estos centros por medio de líneas telefónicas cuya transmisión era de 56,000 bits por segundo (56k bps). Para que todos las Universidades y sus Campus pudieran pertenecer a ésta red, se crearon redes regionales. En cada área del país las escuelas se conectaban a su vecina más cercana y a su vez estas cadenas se conectaban a uno de los centros de cómputo que también estaban conectados entre sí. Dicha configuración hacía factible la comunicación entre una computadora y otra, transmitiendo la información a través de la cadenas.

Debido al éxito de NSFNET, el tráfico de información sobre la red se llegó a incrementar hasta el punto de saturar las líneas telefónicas. En 1987 se contrató a la corporación Merit Network Inc. para reemplazar las computadoras y las líneas telefónicas por otras con mayor capacidad y velocidad de transmisión.

La NSF puso a disposición de cualquier institución escolar, de investigadores académicos, científicos, empleados de gobierno y cualquier otra organización que lo requiriera , la oportunidad de conectarse y hacer uso de la red. Para 1990 Internet habría tomado un gran auge y desde entonces la red está disponible a todo aquel que tenga intenciones de pertenecer a este fenómeno. Todo esto apunta a un crecimiento continuo, nuevos problemas por resolver, demanda de tecnología que asegure la comunicación y el trabajo de los usuarios.

El número de usuarios ha crecido desde casi los 5,000 hasta más de 30 millones en tan sólo 10 años. Es un incremento del 6000% y esto es solo el inicio.

1.2.- MODO DE OPERACIÓN

Al inicio Internet estaba constituida por redes basadas en el protocolo IP, pero tiempo más tarde, organizaciones con redes no basadas en este protocolo, vieron que era un buen servicio y debían proporcionárselo a sus clientes. Tuvieron que desarrollar métodos para conectar esas redes "diferentes" (como BITNET, DECNET, etc.) a Internet. Al principio estas conexiones llamadas "gateways" [2] simplemente servían para transferir correo electrónico entre estas redes y de algún modo hubo quienes lograron intercambiar otro tipo de servicios.

Todas las estructuras que conforman a Internet son redes computacionales, todo desde pequeñas redes de área local (LAN) hasta las masivas redes de cobertura amplia (WAN). Todos esos grupos de redes se encuentran conectados a Internet y de este modo entre cada una de ellas. Para cualquiera existe disponibilidad de comunicación, siendo por línea telefónica, líneas de arrendamiento dedicadas, o hasta enlaces vía microondas. Debido a la gran diversidad, no sólo geográfica, sino también en términos de sistemas operativos y plataformas, se requieren protocolos estándares de comunicación para asegurar la compatibilidad entre configuraciones. En el caso de Internet, los protocolos usados se conocen como el Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP).

---

[2] *Un gateway es un dispositivo que permite a redes desiguales y con diferentes protocolos, que establezcan comunicación.*

Las redes que comprenden a Internet están conectadas por computadoras conocidas como "enrutadores"[3] las cuales deben ser capaces de decidir la manera más adecuada y eficiente de transmitir datos a través de diferentes caminos y partes de la gran carretera de información. El Protocolo de Internet (IP) asegura que estos enrutadores conozcan con exactitud a donde enviar esa información, direccionándola en pequeños paquetes de datos. Estos paquetes son cortos y fácilmente se pueden dañar o extraviar durante el viaje; ahí es donde interviene el Protocolo de Control de Transmisión (TCP) que coloca eficazmente el contenido de esos paquetes dentro de un mismo "sobre" asegurado.

En esencia, esto implica que no importa el tipo de computadora que se emplee, ya sea una sencilla Amiga 500 o una poderosa Pentium PC, el usuario podrá conectarse a Internet y participar de lleno en sus servicios y aplicaciones.

1.3.- ESTRUCTURA POLÍTICA

Internet es inmensamente grande en cuanto a sus componentes y la diversidad de estos. Es un sistema que crece y se desarrolla casi orgánicamente. No existe en realidad un presidente, un director, o alguna autoridad en particular que se haga cargo del manejo de la red. Por su parte, las redes que la constituyen tendrán alguna autoridad que las maneje, pero eso es un asunto que concierne a cada ente por separado.

Por nombrar algunas organizaciones mayormente involucradas en la historia de Internet, recordemos la NSF, un grupo de suma influencia en la creación de la red que continua en el reto del desarrollo y avance de la misma. De igual importancia se encuentra la organización americana llamada ANS (Advanced Network and Services), la cual ha proporcionado gran parte de la infraestructura del sistema y está respaldada por 3 corporaciones de Estados Unidos - Merit (Michigan Education and Research Infraestructura Triad), IBM y el poderoso de telecomunicaciones MCI.

Sin embargo, la forma más directa de control, proviene de aquellos grupos como la Sociedad de Internet (ISOC) en Estados Unidos y la asociación RARE (Research Associes pour la Recherche Europeene) en Europa. Estos dos grupos se basan más en la cooperación voluntaria que en el fomento comercial, y son asociaciones que parecen funcionar bien con la Red.

---

[3] *Se denomina enrutador al sistema que transfiere datos entre redes diferentes que usan los mismos protocolos.*

ISOC es una sociedad de miembros voluntarios con el propósito de promover el intercambio de información a través de la tecnología de Internet. Aquí se designa un consejo de conocedores que tienen la responsabilidad del manejo técnico y dirección de la Red. Este consejo es un grupo de voluntarios invitados a formar parte del IAB (Base de la Arquitectura de Internet). Los miembros del IAB se reúnen regularmente para establecer estándares y asignar recursos tales como direcciones. Internet trabaja gracias a modelos estándares que permiten a las computadoras y aplicaciones de software comunicarse entre sí, esto produce que computadoras de diferentes marcas puedan establecer comunicación sin problemas, y que la red no sea exclusiva de IBM, de Sun o de Macintosh. Este consejo decide cuando son requeridos nuevos estándares, estudia el caso, los habilita y los pone a disposición por medio de la Red a través de documentos conocidos como RFC's[4] (Requests for Comments). También guarda registros de varios números que deben existir como únicos; por ejemplo, cada computadora en Internet tiene una dirección de 32 bits que debe ser única, es decir, ninguna otra máquina tendrá la misma dirección. El IAB no es precisamente quién asigna esos números de dirección, pero si hace las reglas para su asignación.

Existe otra organización voluntaria llamada IETF -Internet Engineering Task Force (Fuerza de Tareas de Ingeniería en Internet) - que regularmente discuten sobre problemas técnicos y de operación de Internet. Si una red acepta la filosofía de Internet y se conecta a ella, prácticamente es parte de la misma, pero puede encontrar ciertas cosas que no le agraden y mandar sus comentarios y sugerencias al IETF. Algunos de estos asuntos se considerarán válidos e Internet podrá modificarlos; de la misma forma si una red realiza cualquier hecho que dañe a Internet, esta podría ser desconectada hasta que remendara los daños ocasionados.

Por último, es importante mencionar que nadie paga a Internet, no existe una compañía que colecte honorarios o cuotas de todas las redes o usuarios de la Red. En lugar de eso, cada quien paga por su parte a la región que pertenezca. La NSF paga los gastos de la NSFNET, la NASA paga por "NASA Science Internet", etc. Las redes deciden como enlazarse entre ellas y consolidan sus interconexiones. Un colegio o corporación paga por su conexión a alguna red regional, que en su turno paga a su proveedor nacional para su acceso.

---

[4] *RFCs son documentos que definen a Internet. Especifican como funciona, como usarla y cuales son sus estándares. La mayoría son de contenido técnico y existen más de 1200.*
Estos documentos pueden ser accesados en algunos servidores vía ftp (p.e. nic.ddn.mil) login:anonymous; cd rfc

En Internet cada red tiene su propio Centro de Operaciones de Red (NOC), estos centros se comunican con cada una de ellas y saben como resolver sus problemas. Su trabajo es mantener un sistema estable y funcional , y cuando las cosas no anden bien, ellos son los encargados de resolverlo.

1.4.- EVOLUCIÓN DEL SISTEMA

Internet ha sido una red Internacional desde hace mucho tiempo. Actualmente interconecta a más de 50 países y el número se incrementa rápidamente. Los países tercermundistas que no tenían intenciones de participar en el enlace, ahora han visto en él un buen medio de incrementar sus niveles de educación y tecnología.

En Europa, el desarrollo de Internet era impedido por las políticas nacionales de los protocolos OSI [5], considerando a IP como una amenaza cultural, excepto por los escandinavos que lo aceptaron mucho tiempo antes y ya estaban bien conectados a ella. En 1989, la organización RIPE (Reseaux IP Europeens) comenzó a coordinar la operación de Internet en Europa, y para este tiempo, cerca del 25% de los hosts conectados a la red, se encuentran en el viejo continente.

Actualmente la expansión internacional de Internet se ve afectada por la carencia de una buena infraestructura de soporte, o mejor dicho, un sistema telefónico con buen rendimiento . Tanto en Europa Oriental como en los países del tercer mundo, tal calidad no existe. Aún en las grandes ciudades, las conexiones están limitadas a las velocidades disponibles que en promedio apenas alcanza los 9600 bits/segundo, que cualquiera rebasa en los Estados Unidos. Por ello sólo algunos lugares pueden accesar a Internet, generalmente las universidades más importantes de cada país. No obstante, tan pronto como la funcionalidad de estos sistemas mejore, muchos otros lugares de menos renombre, podrán conectarse a la Carretera de Información.

1.5.- INTERNET EN MÉXICO : PROVEEDORES DEL SERVICIO

Para conectarse a Internet hay que adquirir los servicios de un proveedor. En México existen por lo menos 16 empresas que ya ofrecen este servicio ,entre ellas se encuentran: Datanet, PixelNet, Connect, SPIN, Infotec, Internet de México, Ashton Communications, CompuServe, Value Added Networks, RedUNAM (cuyo proveedor directo es Advanced Network y Rice University[6]) y otros

[5] *La Organización de Estándares Internacionales (ISO) finalmente diseño su modelo internacional de comunicación y lo nombró ISO/OSI (Open Sytems Interconnect)*
[6] *Información disponible en http://serpiente.dgsca.unam.mx/rectoria/htm/redunam.html*

más. Es recomendable seleccionar a un proveedor evaluando sus tipos de conexión y servicios. Los tipos de conexión más comunes son:

a) Dial-up o por marcación. Es el tipo de conexión más común y sencillo. El único equipo requerido es un modem y un programa de emulación de terminal.
b) IP dial-up o por marcación con dirección IP. En este tipo de conexión se requiere que el equipo de cómputo tenga una dirección IP legal proporcionada por el proveedor de servicios Internet.
c) Conexión directa. Este sistema es caro pero eficiente. Una conexión directa esta orientada principalmente para hacer uso de Internet como el cimiento principal de la empresa.

1.5.1.- REDUNAM

Entre los proveedores de Internet en México, RedUNAM tiene una gran demanda. Esta red es un proyecto creado por la Universidad Nacional Autónoma de México y es una red de comunicación de datos que permite el intercambio de información entre organizaciones académicas (facultades, institutos, centros de difusión, coordinaciones y demás dependencias que la conforman) y de investigación a nivel local, nacional e internacional, a través de conexiones con otras redes externas a RedUNAM.

Una de las políticas que maneja esta red es que, todo tráfico originado por instituciones u organizaciones deben servir o dar soporte a :

- la investigación
- academia
- asuntos de gobierno local, estatal o nacional
- desarrollo económico o de servicio público

A está red se encuentran conectadas otras instituciones educativas, de investigación y comerciales, como son:

- U. Iberoamericana
- U. La Salle
- ITAM
- U. Autónoma Metropolitana
- Colegio de México
- CONACYT
- CIMMYT
- CENIDS
- CINVESTAV
- Centro Nacional para la Prevención de Desastres (CENAPRED)
- IMP
- Consorcio Red Uno

- U. Tecnológica de Nezahualcoyotl
- IIE (Cuernavaca, Morelos)
- CICESE (Ensenada, Baja California)
- U. de Guadalajara
- U. de Guanajuato
- Advanced Network & Systems, Inc. (Houston, Texas)
- Rice University (Houston, Texas)
- National Center for Atmospheric Research (Boulder, Colorado)

Los objetivos que persigue RedUNAM son principalmente: promover el intercambio de ideas, apoyar el crecimiento de la UNAM, y acercar los bancos de información y conocimiento a todo aquel que lo requiera.

Estructura de RedUNAM

RedUNAM está estructurada en forma de anillo de FDDI (fibra óptica activa y una de respaldo que pueden transportar información hasta 100Mbps) y que conecta a 5 enrutadores principales. Conectadas a ellos se encuentran las redes locales de cada dependencia: las que se encuentran en C.U. se enlazan a través de fibra óptica y las que se hallan fuera de él se conectan a la red por medio de alguna de las siguientes vías:

- En el área metropolitana:
  -Radio modem
  -Líneas conmutadas o privadas
  -Microondas
  -RDI (Red Digital Integrada)
- En el resto del país:
  -RDI (Red Digital Integrada)
  -Enlaces vía satélite

En las redes que pertenecen a RedUNAM las topologías más empleadas son variantes de Ethernet: primero aparecen las redes tipo estrella (o red de par trenzado, pues se construyen con este medio físico), también se encuentran aquellas que se conectan a través de coaxial delgado y otras de Token Ring que se encuentran en franca desaparición.

Los protocolos y sistemas operativos que requiere esta red deben:

- Permitir la conexión entre diferentes tipos de computadoras y sistemas operativos que se utilizan en las redes locales

- Operar en condiciones fiables y brindar a los administradores herramientas de monitoreo y mantenimiento preventivo del funcionamiento de la red
- Atención apropiada para redes de área amplia o metropolitana y las de área local.

Los protocolos TCP/IP señalan ser los más viables para dar solución a todo esto. Además es el protocolo de Internet sobre el cual pueden instalarse sistemas operativos de red tales como Windows NT y sus variantes, LAN Manager, Lantastic, etc. así como Netware.

Servicios y Conexión en RedUNAM

RedUNAM permite la comunicación entre los diversos hosts (máquinas que fungen como servidores o clientes) y ofrece servicios como:

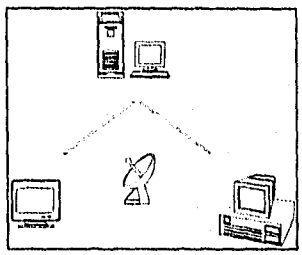
- -Enrutamiento que se encarga de que los archivos no se extravíen o le lleguen a la persona equivocada y puede limitar el acceso de usuarios indeseables.
- -DNS (Domain Name Service - Servicio de Nombres de Dominio) que resuelven la conversión entre las direcciones lógicas y los nombres de las máquinas, de forma que el usuario pueda usar un nombre en lugar de números.
- -NOC/NIC (Centro de Operación de la Red y Centro de Información de la Red) que atienden el monitoreo y mantenimiento de la red, así como las asesorías requeridas por los usuarios.

En base a estos servicios el usuario puede accesar a otros, tales como: Servidores de correo, Gophers (menús jerárquicos que permiten buscar información), Archie (búsqueda de temas en particular), Telnet (sesiones remotas), entre otros.

Dentro de la UNAM se ofrecen diversas formas de conexión a la red general. Una de las más comunes en los Campus de CU, Preparatorias, Colegios de Ciencias y Humanidades, y Escuelas de Estudios Profesionales, es la conexión directa mediante cableado de cobre, par torcido ó fibra óptica. Esta conexión directa llega hasta la computadora que se desea emplear y que trabaja con un software especial de comunicación de red.

Otra manera para los usuarios que desean conectarse fuera del lugar de trabajo es vía modem. Esto permite que mediante un modem una computadora con un programa de emulación de terminal, pueda entrar a la red para emplear los equipos TCP/IP. Para este servicio es necesario obtener una cuenta y password en las oficinas de la DGSCA (Dirección General de Servicios de Cómputo Académico) en Ciudad Universitaria.

1.6.- ELEMENTOS DEL SISTEMA DE INTERNET



El sistema de Internet está constituido de computadoras-hosts conectados a redes que a su vez están interconectadas mediante gateways o enrutadores. Las redes pueden ser tanto de área local (p.ej. Ethernet) o de área extensa (p.ej. ARPANET), pero en ambos casos basadas en una tecnología de conmutación de paquetes (packets switching). Diversos niveles de protocolos (TCP/IP) en las redes, los gateways, y los hosts, soportan un sistema de comunicación de interprocesos que provee un flujo bidireccional en las conexiones lógicas entre los puertos de los procesos.

Los datos son transmitidos de host a host mediante una serie de caminos en un conjunto de redes PSN (packet switched network). Los hosts son computadoras enlazadas a una red, y desde el punto de vista de la red de comunicaciones, son los transmisores y receptores de paquetes de información. Los procesos son vistos como elementos activos en las computadoras (descritos como programas en ejecución). Aún las terminales y archivos y otros dispositivos de entrada y salida (I/O), son considerados como comunicadores por medio del uso de procesos.

Para llevarse a cabo la transmisión de datos es necesario establecer una línea de comunicación. La línea de comunicación se obtiene mediante la utilización de líneas telefónicas, cables, satélites, etc., y pueden ser tres las formas de transmisión de datos:

Simplex.- Los datos se transmiten en una sola dirección. Su principal aplicación consiste en incrementar el flujo de información al no establecerse ninguna interrupción en la transmisión.

Half Duplex.- Los datos se pueden transmitir en dos direcciones pero no de manera simultánea, es decir, solo una a la vez. Su aplicación es frecuente en las operaciones de tiempo compartido.

Full Duplex.- Es el tipo más práctico y de mayor uso, ya que permite que la transmisión se pueda llevar a cabo en las dos direcciones de manera simultánea.

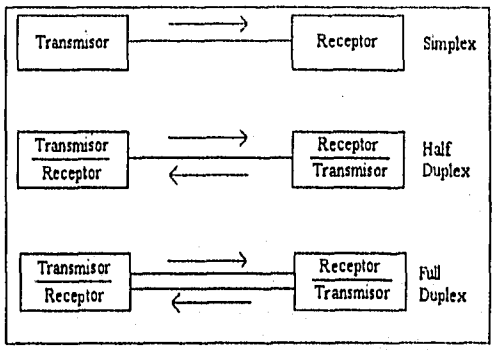


fig. 1.1.- Formas de Transmisión

Para establecer una transmisión de señales entre dos puntos es necesario considerar varios aspectos que involucran diversos conceptos:

- - Velocidad de transmisión
- - Conversión del tipo de señal
- - Los medios de transmisión

La velocidad de transmisión es un elemento que se debe tomar en cuenta para la comunicación y varía dependiendo del medio físico que se utilice para realizarla. La velocidad de transmisión se mide por el número de bits que se transmiten en un segundo (baudios por segundo). Cabe aclarar que ese número de "*baudios por segundo*" no corresponden únicamente a bits de datos, pues generalmente por cada byte de datos se transmiten de 2 a 3 bits de control. Lo que implica que por cada byte de datos se transmitan de 10 a 11 bits.

La velocidad de transmisión se clasifica en tres grupos principales conocidos como "*anchos de banda*":

Canal de banda angosta.- El rango de velocidades varía desde 45 hasta 150 bps (baudios por segundo) y corresponden a las velocidades de transmisión más lentas. Se aplican en líneas telegráficas principalmente.

Canal de Banda de Voz.- En este canal se considera un rango de velocidades comprendido entre 1800 y 9600 bps. Las transmisiones en banda de voz se utilizan frecuentemente dado que muchas de ellas se llevan a cabo a través de líneas telefónicas.

Canal de Banda Ancha: Esta banda posee las velocidades más altas, siendo estas desde 19200 bps. Las transmisiones se efectúan en medios como lo son los cables coaxiales, fibras ópticas y microondas.

La mayoría de los medios que se emplean para transmitir señales (como las líneas telefónicas), transportan la señal de manera analógica y debemos recordar que casi todas las computadoras manejan la información en forma digital, por lo que surge la necesidad de realizar una conversión del tipo de señal, de tal manera que los datos se conviertan del tipo digital al tipo analógico para viajar en el medio de comunicación. A su vez también es necesario convertir una señal analógica a una de tipo digital para recibir una transmisión de datos y posteriormente procesarlos. El dispositivo que se encarga de realizar esta función se denomina *modem*. En principio el modem tiene la capacidad de convertir el código binario en una señal analógica denominado *modulación* y de manera opuesta definido como *demodulación*.

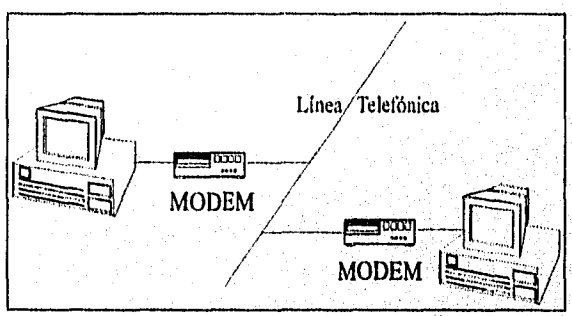


fig. 1.2.- El uso de modems en la transmisión

Los medios de transmisión son los elementos que hacen posible el envío de señales. Estos medios se encuentran en constante evolución y cada avance trae consigo innovaciones sorprendentes, principalmente en la capacidad de transmisión. Entre los medios de transmisión más comunes se encuentran:

- Línea de Transmisión telefónica
- Cable Coaxial.- El cable coaxial es un cilindro hueco de cobre llamado blindaje que rodea a un alambre con aislante. Esta estructura se diseñó para proteger al conductor de cualquier interferencia provocada por el medio ambiente para obtener así una transmisión clara y a una velocidad más alta.

- Transmisión de microondas.- La transmisión a través de microondas consiste en señales electromagnéticas que viajan a través de la atmósfera y el espacio. Las velocidades de transmisión son altas, manejan señales para sonido e imagen. Las microondas son transmitidas básicamente por dos tipos de mecanismos:
  - Torres de Transmisión: Se conocen también como Estaciones de Microondas y deben estar en línea visual unas de otras a distancias no mayores de 40 a 50 kilómetros.
  - Satélites: Mediante el empleo de satélites artificiales se pueden realizar comunicaciones intercontinentales ; para esto es necesario colocar al satélite a una altura aproximada de 37,000 Km para que este gire a la misma velocidad con la que rota la Tierra y de esta manera se convierta en una estación fija de transmisión de microondas.
- Fibras ópticas: Las fibras ópticas constituyen uno de los avances tecnológicos más recientes. Construidos a base de hilos delgados y flexibles por donde puede viajar en su interior un rayo láser. Actualmente las fibras ópticas están sustituyendo a los cables coaxiales ya que pueden transmitir 100,000 veces más información de la que se puede transmitir por medio de microondas.

Las compuertas requeridas para unir las diversas redes que pertenecen a Internet son conocidas como enrutadores y/o gateways. Un *enrutador* es un dispositivo que se encarga de "enrutar" (llevar por la mejor ruta) la información entre redes. Aunque un enrutador puede transferir datos entre redes que utilicen la misma tecnología, generalmente su empleo se debe a su capacidad de transferir datos entre redes con tecnologías diferentes como Ethernet y Token Ring, que usan protocolos iguales. Por su parte, los *gateways* permiten la transferencia de datos entre diferentes tipos de redes que emplean diferentes protocolos.

Debido a que Internet consta de miles de redes que usan diversas tecnologías, estos elementos son parte esencial de la misma. Un enrutador o un gateway tienen una dirección dentro de la red, pues son usados como destinos intermediarios.

La comunicación entre redes requiere una conexión entre dos computadoras o programas que intercambien información una con la otra. Una conexión de red consiste de 2 extremos en el proceso de comunicación, el cliente y el servidor. El extremo cliente es aquel lado de la conexión que está solicitando información o servicios al otro extremo de la conexión (servidor). El lado del servidor responde a las peticiones del cliente. Esto implica que las aplicaciones de una red desarrollan 2 funciones: solicitud de información y atención a la solicitud de información. El programa que realiza la función de petición es un programa cliente y aquel programa que responda a dichas peticiones funciona como un programa servidor.

Los procesos necesitan distinguirse de otros procesos en medio de varios flujos de comunicación, y para esto se supone que cada proceso tenga un número de identificación (puertos) a través de los cuales se comunique con los puertos de otros procesos.

En los siguientes capítulos analizaremos estos términos con mayor amplitud.

Es el tiempo de "viajar" a través de esta Super Carretera de Información, conocerla y que cada quien tome su camino.

Capítulo II
Protocolos TCP/IP

Antes de entrar directamente en el tema de los protocolos de Internet (TCP/IP) exploremos algunos conceptos útiles para una mejor comprensión.

Sabemos ya que dos ó más computadoras conectadas a través de un canal de comunicación forman una red. Para transferir datos entre computadoras las redes utilizan lo que se conoce como una comunicación conmutada . Este método permite que diversos dispositivos de hardware compartan líneas físicas de comunicación (y evitar que cada computadora tuviera las *n* líneas para comunicarse con las *n* computadoras de toda la red). Los dos métodos más comunes de conmutación en la comunicación son: conmutación de circuitos y conmutación de paquetes.

La conmutación de circuitos crea una sola e irrompible ruta entre dos dispositivos que establecen comunicación. Mientras esos dos dispositivos se comunican, ningún otro elemento podrá usar esa ruta. Sin embargo, cuando los dispositivos hayan terminado de transferir información, liberan ese camino de comunicación y así otros elementos podrán usarla. En otras palabras, el método de conmutación por circuitos deja que los diferentes elementos compartan las líneas de comunicación, pero cada uno deberá esperar su turno.

Por otro lado, una red con conmutación de paquetes divide la información en fragmentos ó paquetes para transmitirlos entre computadoras. Estos paquetes contendrán una dirección de destino y un número de secuencia para que la computadora que los reciba pueda ordenarlos y volver a construir el mensaje original . Las redes que conmutan paquetes consisten de 2 componentes básicos: los elementos conmutadores (switches) y las líneas de transmisión.

Los elementos conmutadores son aquellos capaces de ayudar a los paquetes para encontrar y alcanzar su destino. Estos elementos existen en varios puntos de las redes, que al cambiar de ruta (como en las vías de un tren), se puede controlar el destino de cada paquete que transite por las líneas de transmisión. Entre estos conmutadores encontraremos desde una computadora, puentes, enrutadores, gateways, o cualquier otro que desarrolle funciones similares.

Un puente es un dispositivo que conecta redes que usan la misma tecnología (como Ethernet). Un enrutador transfiere o enruta datos entre computadoras de redes con tecnologías diferentes (Ethernet, ARCnet, Token Ring, ATM, etc.) con los mismos protocolos, y los gateways entre redes desiguales con diferentes protocolos. Los 2 últimos tienen una dirección dentro de la red y un puente no.

2.1.- MODELO OSI

Los diseñadores de redes se basan en ciertos principios que rigen la forma en que deben colocarse todos los componentes y poder estructurar una red.

Para poder entender la comunicación entre pares de computadoras pertenecientes a una red, es indispensable conocer los principios fundamentales del modelo estándar OSI.

Ya se ha dicho que una de las principales características de las redes que conforman a Internet es el hecho de incorporar una gran variedad de dispositivos diferentes, cada uno de los cuales puede ser fabricado por diversas compañías. Para lograr esta compatibilidad de comunicación tanto en software como en hardware han sido creados los estándares o modelos que definen las diferentes funciones a seguir por cualquier dispositivo que pertenezca a una red.

El modelo *OSI* (Open Systems Interconnection- Conexión de Sistemas Abiertos) divide esas funciones y las organiza en 7 capas funcionales donde cada una es responsable de ejecutar ciertas tareas especificas y de proporcionar determinados servicios. Cada una de las capas ofrece una serie de servicios a la capa superior y utiliza los servicios que le brinda la capa inferior. Los servicios de una capa pueden verse como una interfaz de comunicación con la misma. Los protocolos por su parte, definen la forma que van a tener las tramas de bits que intercambian las distintas capas y como se va a llevar a cabo esa comunicación.

El objetivo de separar protocolos de servicios es el de aislar los aspectos tecnológicos de los aspectos de uso de una red. De esta forma, se definen servicios lo menos cambiantes posible para que los usuarios no tengan que estar modificando sus aplicaciones continuamente. Por otro lado, las mejoras en las tecnologías de las redes van a repercutir en diseño de protocolos que garanticen una transmisión más fiable, segura y rápida. Esto afecta a los protocolos, pero no a los servicios.

A continuación la fig. 2.1. ilustra la estructura en capas del modelo OSI para la comunicación entre computadoras.

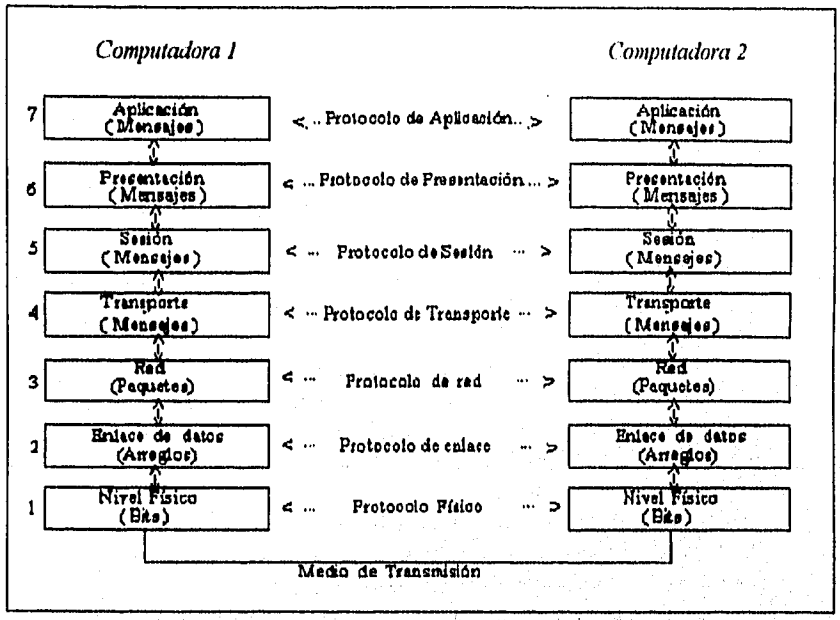


fig. 2.1 Capas funcionales del Modelo OSI

Las 7 capas que estructura este modelo son:

1.- *Capa física*: Se encarga de la transmisión de los bits a lo largo de un canal físico de comunicación. Esta capa incluye todo el hardware requerido para llevar a cabo la transmisión.

2.- *Capa de enlace*: La capa física maneja datos en bruto como bits (1's y 0's). La capa de enlace transforma esos datos en algo comprensible, normalmente en arreglos de bits, para la capa de red. Esta capa de enlace también acepta información del nivel de red y lo traduce al formato binario que usa la capa física. La tarea primordial de esta capa es la de detectar y prevenir la alteración de los datos en el canal de comunicación. La tarjeta interfaz de red (Ethernet, Token Ring, ARCnet) que sirve para conectar una computadora a una red, representa y provee las funciones de la capa de enlace.

3.- *Capa de Red*: En una red de conmutación de paquetes como Internet, esta capa se ocupa de la obtención de paquetes de datos procedentes de la fuente y de encaminarlos por la red y subredes hasta alcanzar su destino. Cada paquete de información debe contener una dirección de origen y una dirección de destino para facilitar el proceso de entrega. Esta capa de red asegura que los hosts reciban los paquetes correctos y en la secuencia original.

4.- *Capa de Transporte*: Cuando dos computadoras se comunican a través de la red, en realidad son 2 procesos los que intercambian datos. La capa de red se encarga de entregar los datos a la capa de transporte de una computadora y esta capa a su vez entregará la información al programa o aplicación correspondientes. En una red de conmutación de paquetes, la capa de transporte deberá romper o fragmentar la información que recibe de la capa de sesión en pequeños paquetes que la capa de red requiera. Por otra parte, en el extremo receptor, la capa de transporte debe reensamblar la información fragmentada. Finalmente esta capa produce el tráfico de paquetes que la capa de red debe manejar.

5.- *Capa de Sesión*: Permite que usuarios de diferentes máquinas puedan establecer sesiones de trabajo en máquinas remotas. Esta capa negocia las conexiones entre procesos o aplicaciones en diferentes computadoras de la red. La capa de sesión es la interfaz del usuario con la red.

6.- *Capa de Presentación*: La capa de presentación determina como aparecerá la información al usuario. Contiene funciones que sirven como interfaz de la red con diversos tipos de impresoras, monitores y formatos de archivos. Esta capa esconde las diferencias de los dispositivos de hardware que pueden afectar la forma en que la red despliega, imprime o interpreta los datos para un usuario.

7.- *Capa de Aplicación*: La capa de Aplicación contiene todos los detalles relativos a las aplicaciones o programas específicos diseñados para los usuarios de red. Esta capa resuelve el problema de compatibilización de los distintos terminales que hay en el mercado, con objeto de que los programas de aplicación no tengan que preocuparse de conocer las características del hardware de los mismos.

Hemos visto que en el diseño de redes se utilizan capas para organizar las comunicaciones de una red en módulos funcionales bien definidos. Estas capas conforman un modelo del sistema de comunicación de las redes y este sistema también incluye protocolos que definen reglas y convenciones para la comunicación entre cada una de las capas del modelo.

Los protocolos son reglas que definen como debe ser el trabajo del Software. Los Sistemas Operativos utilizan estas reglas (protocolos) para el manejo de información que circula entre usuarios, sus aplicaciones y las computadoras. De esta forma, *los protocolos controlan el flujo de la información entre las computadoras y los programas de la red.*

Dicho de otra forma, los protocolos señalan de que manera colocar los bits en un medio de transmisión y enviarlos con un formato determinado. Los protocolos estándares permiten que computadoras diferentes en hardware y software logren comunicarse e intercambiar información.

2.2.- PROTOCOLOS *TCP/IP*

La Super Carretera de Información, Internet, cuenta con una colección de protocolos conocida como la serie de protocolos TCP/IP . Esta serie incluye los siguientes protocolos los cuales trabajan en conjunto en el intercambio de información a través de Internet:

PROTOCOLO	PROPÓSITO
IP	El Protocolo de Internet es un protocolo perteneciente a la capa de red y se encarga de mover información entre computadoras.
TCP	El Protocolo de Control de Transporte pertenece a la capa de transporte de una red y traslada información entre aplicaciones.
UDP	El Protocolo de Datagramas del Usuario es otro protocolo de la capa de transporte. UDP también intercambia mensajes entre aplicaciones, pero es menos complejo y fiable que TCP.
ICMP	El Protocolo de Mensajes de Control de Internet acarrea mensajes de error en la red y reporta otras condiciones que requieren atención del software de la red.

Sabemos que el modelo OSI divide en 7 capas la estructura de comunicación de una computadora en una red. Cada capa asocia protocolos que definen su funcionalidad. En la figura 2.2 se ilustran las capas de este modelo y su correspondiente relación con los protocolos TCP/IP.

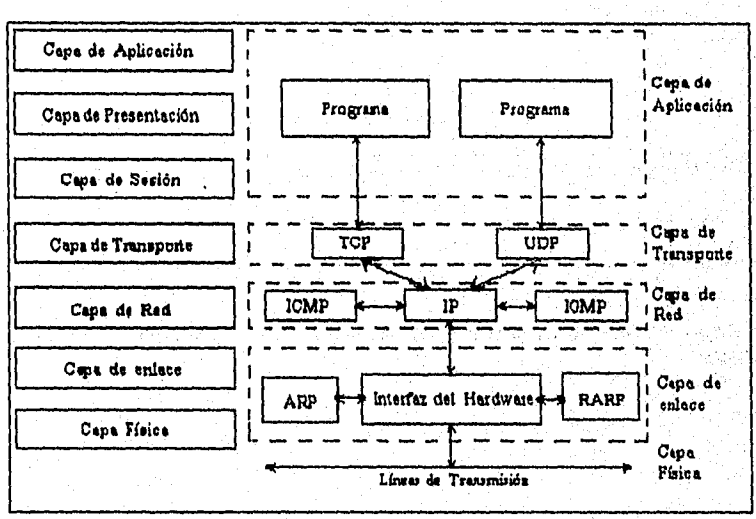


fig. 2.2 Modelo OSI y la pila de protocolos TCP/IP

El termino pila de protocolos se refiere a la combinación de las capas de la red y sus protocolos correspondientes. Cuando se transmite información a una máquina remota, la información local fluye de arriba hacia abajo por la pila de los protocolos (en las capas del modelo OSI) y cruza la red a través de las lineas de transmisión. Una vez en su destino, la información fluye ascendentemente por la pila de protocolos hasta alcanzar el programa correspondiente en la máquina remota.

En la siguiente tabla se resumen las caracteristicas de la función de cada uno de los protocolos en la serie TCP/IP:

Protocolo	*Características*
IP (Internet Protocol) *Capa de Red*	Direccionamiento servidor-a-servidor Enrutamiento Transmisión por paquetes, fragmentación y desfragmentación.
UDP (User Datagram Protocol) *Capa de Transporte*	Inseguro (no garantiza la entrega de la información) Muy Simple Direccionamiento punto-a-punto (peer to peer) Verificaciones Opcionales
TCP (Transport Control Protocol) *Capa de Transporte*	Transporte seguro y confiable (paquetes en orden, induplicados, etc.) Más Complejo Control de flujo Indicación de datos urgentes
ICMP (Internet Control Message Protocol) *Capa de Red*	Protocolo para mensajes de control en la red de Internet.

Ahora describiremos un poco más las características de los protocolos y sus funciones primordiales.

2.2.1.- PROTOCOLO DE INTERNET (IP)

Como pudimos observar en la figura 2.2, la capa de red incluye los módulos del Protocolo de Internet (IP), el Protocolo para Mensajes de Control de Internet (ICMP), y el Protocolo para el Manejo de Grupo de Internet (IGMP). Dentro de esta capa de red, IP es el que ejecuta la mayoría del trabajo. ICMP y el IGMP son protocolos soportados por IP y que lo ayudan en el manejo de mensajes especiales de red como errores y de "multicast" (mensajes enviados a 2 ó más sistemas).

El Protocolo de Internet es el sistema de entrega de la serie de protocolos TCP/IP. Los protocolos TCP, UDP, y hasta ICMP necesitan del protocolo IP para la entrega de información.

El Protocolo IP se denomina como un protocolo *servidor-a-servidor*, ya que envía datagramas desde un host de una red local hasta entregarlos en el host de destino.

El Protocolo IP utiliza un método "no-fiable" en la entrega de información a través de "datagramas IP" (paquetes enviados en una red de conmutación de paquetes) que cruzan la red . Las redes TCP/IP transmiten toda la información a través de Internet mediante datagramas IP. Cada datagrama incluye un encabezado IP y los datos reales.

Un protocolo que usa datagramas, transmite la información como un paquete individual de unidades de información. En otras palabras, el protocolo transmite cada datagrama de forma independiente - el datagrama no depende de ningún otro datagrama. De esta forma, múltiples datagramas que el protocolo envía a un mismo destinatario, tal vez no lleguen en el mismo orden que la secuencia de transmisión. Si la aplicación receptora requiere información secuencial , la aplicación debe de cotejar los datos después de su llegada. El protocolo IP y UDP utilizan datagramas para entregar información.

El método de agregar un encabezado a los datos recibidos y encapsularlos dentro de un nuevo paquete crea un datagrama IP que incluye un encabezado IP y la información. El software de la red siempre crea un encabezado IP en múltiplos de palabras de 32 bits. Este encabezado incluye toda la información necesaria para entregar la información encapsulada dentro del datagrama IP.

Debido a que IP es el sistema de entrega de todo Internet, un encabezado IP contiene gran cantidad de información. Sin embargo, a pesar de su gran importancia y de la cantidad de información que este contiene, un encabezado IP solo consume 20 bytes (5 palabras de 32 bits) de espacio almacenado. La siguiente figura nos muestra un datagrama IP con los campos que identifican al encabezado IP.

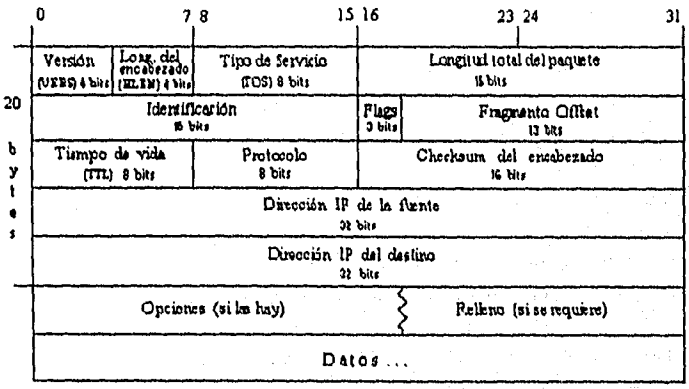


fig. 2.3 Estructura de un datagrama IP

Aunque la figura 2.3 muestra la estructura del encabezado en forma de capas, se debe entender que el encabezado es simplemente una corriente de datos de por lo menos 20 bytes de longitud. Los siguientes párrafos dan una breve descripción de cada campo del encabezado IP.

Número de Versión (VERS)

Los primeros 4 bits del encabezado identifican la versión del Protocolo de Internet utilizado para crear el datagrama.

Longitud del Encabezado (HLEN)

Los siguientes 4 bits especifican la longitud del encabezado IP en palabras de 32 bits. Si no existen opciones especiales, un encabezado IP tendrá una longitud de 20 bytes. Debido a que los datos encapsulados comienzan después del encabezado, por medio de este campo, los protocolos y aplicaciones pueden determinar con exactitud donde se encuentra la información original

Tipo de Servicio

Los próximos 8 bits definen las prioridades para un paquete IP. El campo de tipo de servicio en el encabezado IP informa a la capa de red sobre decisiones del manejo relacionadas con las prioridades en la entrega de la información.

Longitud del Paquete

El siguiente campo de 16 bits especifica la longitud total del paquete IP, incluyendo el encabezado IP. Utilizando la longitud del encabezado y la del paquete, se puede averiguar el principio y fin de los datos encapsulados. Debido a que la longitud del paquete es un campo de 16 bits, teóricamente el tamaño máximo de un datagrama IP es de 65,535 bytes. Sin embargo, TCP/IP puede realizar futuras encapsulaciones de ese mismo datagrama mediante va pasando la capa de enlace. Por ejemplo, si la red local utiliza tecnología Ethernet, la capa de enlace re-encapsulará los datagramas IP

en arreglos para Ethernet antes de transmitir los datos. Cada tecnología de red especifica un tamaño máximo de paquete MTU (máxima unidad de transferencia) que puede aceptar. Por ejemplo, Ethernet limita su transferencia a 1,500 bytes, IBM Token Ring transfiere como máximo 4,464 bytes, etc.

Si una aplicación trata de transmitir paquetes IP más largos que lo especificado por su MTU, tendrá que hacerse una nueva fragmentación. La fragmentación rompe un datagrama en 2 o más fragmentos para ser enviados en múltiples transferencias.

Identificación

Ya que frecuentemente las redes dividen los datagramas en piezas más pequeñas, los diseñadores de TCP/IP incluyen un campo de identificación en el encabezado IP. Los hosts utilizan este campo de 16 bits para identificar cada datagrama que se envía. Cuando una computadora recibe los datagramas, utiliza el campo de identificación para determinar que fragmentos pertenecen a que datagramas.

Banderas y Fragmento Offset

En general las computadoras que son hosts utilizan los campos de Identificación, Banderas y Fragmento Offset para ensamblar paquetes IP fragmentados.

El campo de Fragmento Offset señala a que parte del datagrama original pertenece un fragmento. Este campo es medido en unidades de 8 bytes (64 bits).

Tiempo de Vida

Este campo de 8 bits determina cuanto tiempo puede un paquete existir fuera de la red. Este tiempo es establecido por el módulo de transmisión y se reduce conforme es llevado a cabo el proceso de enrutamiento (direccionamiento). Si el tiempo de duración llega a cero antes de que el datagrama alcance su destino, este último es destruido. El propósito de este campo es prevenir que los paquetes se pierdan en las carreteras de información.

Protocolo

Hemos visto que la capa de transporte de una red TCP/IP incluye dos protocolos: UDP y TCP. Ambos protocolos requieren de IP para la entrega de información. El campo de Protocolo de 8 bits del encabezado IP indica cual de esos protocolos fue utilizado para crear el encapsulamiento de los datos dentro del paquete. Por ejemplo, si el campo de Protocolo contiene el valor 6 (00000110 en binario), entonces sabrá que el software de la red formateó el área de los datos como un segmento TCP. Si por otro lado, el campo señala el número 17, entonces el área de los datos fue formateada como un datagrama UDP.

La capa de red utiliza el valor del campo de Protocolo cuando transfiere la información ascendentemente en la pila de protocolos mediante la capa de transporte. Al examinar este campo, la capa de red sabe a que módulo de transporte debe contactar.

La tabla posterior indica los valores para los protocolos que usan IP.

Protocolo	Decimal	Binario
ICMP	1	00000001
IGMP	2	00000010
TCP	6	00000110
UDP	17	00010001

Valores del campo de Protocolo del encabezado IP

Suma de Verificación del Encabezado (Checksum)

Los protocolos utilizan checksums para detectar errores en la transmisión de datos. El campo de "Checksum de Encabezado" del encabezado IP contiene un número de 16 bits que representan una suma solamente de los campos del encabezado IP. Este campo no incluye el área de los datos del paquete.

Este chequeo del encabezado verifica que la información sea transmitida correctamente. Si este chequeo falla, el datagrama se descarta inmediatamente desde el módulo que lo detecta.

El protocolo IP no lleva un control de error de los datos, solo un chequeo por encabezado. No hay retransmisiones ni control de flujo.

IP es un protocolo "no-fiable", de ahí que no garantice la entrega. Sin embargo, el checksum del encabezado de IP si garantiza la validez del encabezado del datagrama. Esto es, IP detecta y descarta cualquier paquete que haya sido alterado. No obstante, TCP/IP no requiere que IP reporte el hallazgo de paquetes alterados. Protocolos "fiables" (como TCP) no dependen de IP para reportar esos errores, ya que ellos utilizan su propio mecanismo de detección de errores.

Dirección IP de la Fuente y del Destino

El campo de 32 bits para la dirección de la Fuente contiene la dirección IP (dirección de Internet) del host transmisor (tarjeta interfaz). No importando por cuantos enrutadores tenga que pasar el paquete para alcanzar su destino, este campo nunca cambia. El campo de la dirección IP de la Fuente siempre contiene la dirección del transmisor original. El campo de la dirección IP del Destino contiene una dirección IP estándar de 32 bits. Dependiendo del tipo de mensaje, el campo de la dirección IP del Destino puede contener una dirección IP de un host (unicast[7]) o todos 1's para un mensaje broadcast[8] (más de un host a la vez).

Opciones

El campo de Opciones de IP provee rasgos para controlar la forma en que la red fragmenta y enruta (direcciona) los paquetes IP. Debido a que este campo es para propósitos de prueba y depuración, TCP/IP no requiere que los protocolos de la red almacenen alguna información en el

[7] *Dirección que identifica a un simple host en una red*
[8] *Dirección que engloba a todos los hosts de una red*

campo de Opciones. No todos los hosts y enrutadores soportan todas las Opciones de IP. De cualquier modo, la mayoría del software de red raramente usan las Opciones IP.

Este campo proporciona las funciones de control requeridos en algunas situaciones pero innecesarias para las comunicaciones ordinarias. Estas opciones incluyen provisiones para señales de tiempo, seguridad, y enrutamientos especiales.

Modelo de Operación

Sabemos que en el proceso de comunicación entre dos computadoras la información fluye de arriba hacia abajo en la pila de protocolos TCP/IP en el host transmisor y una vez en el host de destino, esta información cruza las capas de esta pila en forma ascendente (de abajo hacia arriba). Cada una de estas capas de la pila de protocolos empaqueta o encapsula la información en una presentación propia de cada capa. Es decir que cada capa tiene un formato particular para manejar los datos y enviarlos a las capas adyacentes .

La figura 2.4 muestra como cambia el formato de una unidad de datos conforme atraviesa la pila de protocolos TCP/IP .

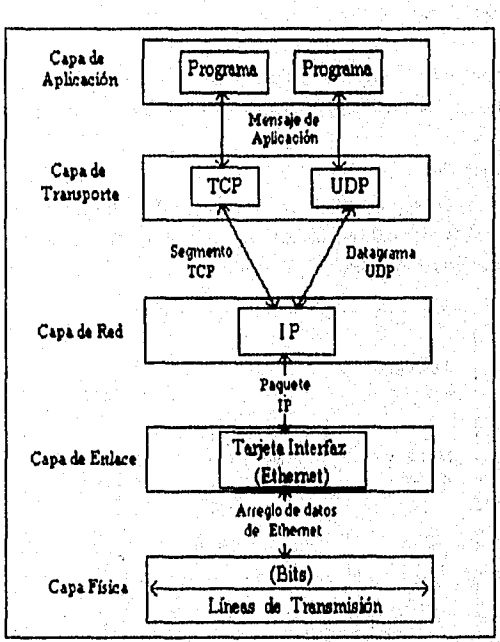


fig. 2.4 Como cambia de formato una unidad de datos conforme atraviesa la pila de protocolos TCP/IP

Podemos hacer referencia a los datos del usuario como mensajes de aplicación cuando estos se mueven entre la capa de aplicación y la capa de transporte. La capa de transporte encapsula esos datos utilizando ya sea el Protocolo de Control de Transporte (TCP) o el Protocolo de Datagramas del Usuario (UDP). TCP utiliza un servicio de entrega mediante "corrientes de bytes" y UDP lo hace mediante datagramas. Dependiendo de cual se emplee, los datos serán referidos respectivamente como segmentos TCP o datagramas UDP conforme estos se trasladan de la capa de transporte hacia la capa de red.

Como se indica en la fig. 2.4, ya sean segmentos TCP o datagramas UDP, ambos se convierten en paquetes o datagramas IP cuando se mueven de la capa de red a la capa de enlace. Una red TCP/IP encapsula todos los segmentos TCP o datagramas UDP dentro de un datagrama IP cuando estos datos viajan de la capa de red hacia la capa de enlace. De manera que se puede denotar a los datos como un datagrama IP o paquete IP cuando estos pasan a la capa de enlace.

Si la red local utiliza tecnología Ethernet, el software de la red encapsulará los datos en arreglos de Ethernet mientras estos se mueven de la capa de enlace hacia la capa o nivel físico. La capa de enlace es donde se encuentra la Tarjeta Interfaz de la Red y de acuerdo a su tecnología (p.ej. Ethernet) estos datos se estructuran ahora en arreglos particulares de cada una de ellas y así los datos están listos para poder viajar a través de las líneas de transmisión.

Por último, podemos observar en la fig. 2.4 que cuando los datos se mueven hacia afuera de la capa de enlace, los datos son referidos como "arreglos de datos", en este caso arreglos de Ethernet.

Ejemplo de Operación

En el caso especial del protocolo IP, el documento RFC 791 "*Internet Protocol*", presenta el siguiente diagrama para ejemplificar brevemente el modelo de operación para transmitir un datagrama desde un programa de aplicación hacia otro:

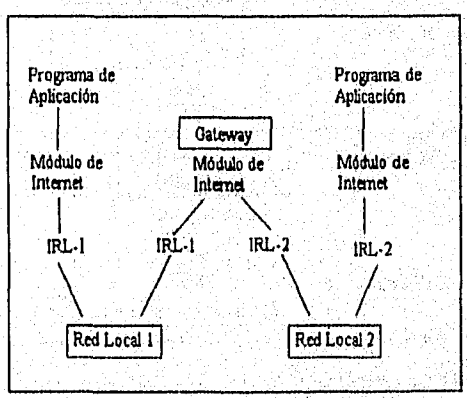


fig. 2.5 Trayectoria de Transmisión

Suponiendo que en la transmisión de un datagrama interviene solamente un gateway como intermediario, a continuación se describe la trayectoria de transmisión que este seguiría :

El programa de aplicación transmisora prepara los datos y los pasa a su módulo local de Internet para que los transmita encapsulados en datagramas IP. La aplicación también le pasa la dirección del host receptor y otros parámetros requeridos en la petición.

Por su parte, el módulo de Internet prepara un encabezado IP y le adjunta los datos de la aplicación. Este módulo de Internet determina una dirección para la red local, que en este caso es la dirección de un gateway, y después envía el datagrama y la dirección de la red local hacia la interfaz de la red.

La interfaz de la red local (IRL) ahora crea un encabezado para la red local y le adjunta el datagrama creado en el módulo de Internet. Una vez que ha creado ese nuevo paquete lo transmite a través de la red.

Ese datagrama llega, encapsulado junto con el encabezado de la red local, a un gateway intermediario. La interfaz local de ese gateway desencapsula este encabezado y envía solamente el datagrama a su módulo de Internet. El módulo de Internet del gateway determina por la dirección IP (contenida en el encabezado del datagrama) que este datagrama deberá transmitirse a otro host en una segunda red. El módulo de Internet determina la dirección IP de la red a la que pertenece el host de destino. Acto seguido, llama a la interfaz de esa red local para enviar el datagrama.

La interfaz de esta red local crea un encabezado específico de red local y le adjunta el datagrama para enviar este nuevo paquete al host de destino.

Una vez en el host de destino, el datagrama es desprendido del encabezado de la red local y mandado al módulo de Internet local.

El módulo de Internet verifica que ese datagrama esté dirigido a un programa de aplicación de ese host y entonces manda los datos a esa aplicación como fue requerido por el sistema transmisor.

Función

El propósito del protocolo IP es mover datagramas a través de un conjunto de redes interconectadas. Como observamos en el ejemplo anterior (fig. 2.5), esto se lleva a cabo transfiriendo los datagramas desde un módulo de Internet a otro hasta que llega a su destino. Los módulos de Internet radican en los hosts y los gateways en el sistema de Internet. Los datagramas son enrutados desde un módulo a otro, cruzando redes individuales, basándose en la interpretación de direcciones de Internet.

En el proceso de enrutamiento de mensajes desde un módulo a otro, los datagramas quizá atraviesen una red cuyo tamaño máximo de paquete sea más pequeño que el tamaño original del datagrama. Para vencer este obstáculo se provee un mecanismo de fragmentación en el módulo de IP.

Direccionamiento

Un nombre nos indica que es lo que buscamos, una dirección señala donde se encuentra y el enrutamiento puntualiza como llegar a ese lugar. El protocolo de Internet opera fundamentalmente con las direcciones. Es la tarea de los protocolos de nivel superior (host-to-host) realizar la traducción de los nombres a sus respectivas direcciones. El módulo de Internet convierte direcciones de Internet a direcciones de redes locales. Los procedimientos del nivel más bajo (gateways o la red local) tienen a su cargo trasladar las direcciones de la redes locales a las rutas.

Las direcciones de Internet están compuestas por 4 bytes (32 bits). Una dirección empieza con un número de red, seguida por la dirección local . Existen 3 formatos ó clases de direcciones de Internet:

Clase A: el bit más alto es 0, los próximos 7 bits especifican la red y los 24 restantes señalan la dirección local.

Clase B: Los 2 bits más altos son 1 y 0, los próximos 14 bits describen la red y los últimos 16 son la dirección local.

Clase C: Los 3 bits más altos son 1 1 y 0, los siguientes 21 bits son para la red y los 8 sobrantes son para la dirección local.

En el capítulo siguiente examinaremos con mayor detenimiento las direcciones IP.

Fragmentación

Sabemos que cada tecnología de red como Ethernet, especifica una unidad máxima de transferencia (MTU). Esta unidad MTU define el tamaño máximo del paquete que la red puede transmitir. Cuando una aplicación transmite un paquete mas grande que lo determinado en su máxima unidad de transferencia, el software de la red rompe automáticamente ese paquete en pedazos más pequeños y transmite la información en múltiples paquetes.

La fragmentación es el proceso de romper un datagrama en dos o más paquetes menores. Este proceso también ocurre cuando un paquete atraviesa por un enrutador y el MTU de este elemento es más pequeño que el MTU de la red local transmisora.

Para el control de la fragmentación, IP utiliza el primer y último bit del campo de Banderas de 3 bits. TCP/IP refiere al primer bit de este campo de Bandera como el bit de "no fragmentar". Si un programa establece el bit de "no fragmentar", pero IP determina que es necesario que se lleve a

cabo la fragmentación del paquete para poder transmitirlo, TCP/IP descartara ese paquete y regresara al transmisor un mensaje de error .

TCP/IP usa al último bit del campo Banderas como la bandera de "más fragmentos". Mediante el proceso del rompimiento de un paquete en piezas más pequeñas, IP activa (pone en 1) esta bandera para cada fragmento que se crea excepto el último de la serie. En otras palabras, esta bandera de "más fragmentos" es verdadera (puesta a 1) para cada fragmento menos el último de ellos. Para el último fragmento, esta bandera es falsa (puesta a 0).

Por razones de eficiencia , IP siempre trata de enviar paquetes lo más grande que se pueda. No obstante, existen ocasiones cuando es imposible evitar la fragmentación de un paquete a través de la red. Para crear cada fragmento, IP calcula un punto de ruptura que resulta en el tamaño del paquete igual al especificado en el MTU de la red. Este punto de ruptura es el byte de ubicación del paquete donde IP lo divide.

El punto de ruptura representa la distancia desde el inicio del datagrama hasta el punto dado. IP almacena cada punto de ruptura en el campo "Offset del Fragmento" en el encabezado del datagrama recién creado .Esto indica que el encabezado del datagrama IP que contendrá el nuevo fragmento de la información, incluirá el valor offset para ese fragmento. Más tarde, en el destinatario final, IP hará uso de este punto de ruptura para reconstruir el paquete.

Desfragmentación

Para reconstruir paquetes fragmentados, el host receptor examina los campos del encabezado IP: Identificación, Banderas y Offset.

Cuando un host recibe un paquete IP con la bandera activa de "más fragmentos" , el host inicializa un contador de tiempo para la reensamblación. Todos los fragmentos deben arribar antes de que el tiempo indicado en ese contador se venza. Si el contador de reensamblación termina antes de que el host reciba todos los fragmentos, el host descarta todo lo que había recibido hasta el momento y no procesa el datagrama. Mientras el host recolecta los fragmentos en su buffer de reensamblación, los módulos de IP utilizan los campos de Dirección de la Fuente y de Identificación para determinar que paquetes pertenecen al mismo datagrama. Cuando el host recibe un fragmento con la bandera de "más fragmentos" desactivada, el módulo de IP puede calcular la longitud del datagrama original.

El campo Offset del fragmento (medido en unidades de 8 bytes) especifica el punto de inicio de ese fragmento medido desde el inicio del paquete original. Es del fragmento final (aquel con la bandera de "más fragmentos" puesta a 0), donde el módulo de IP calcula la longitud del datagrama original agregando los valores de los campos Offset y Longitud del Paquete.

Long. del datagrama original = campo offset del último fragmento + long. de paquete del último fragmento

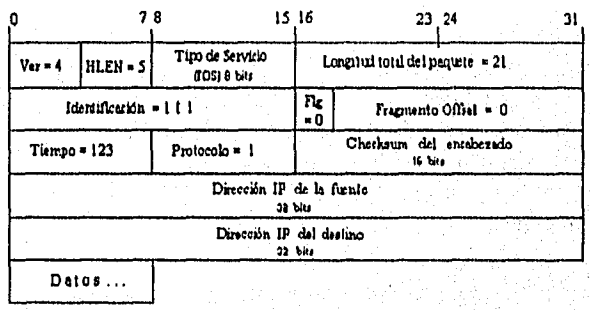
Una vez que el host recibe todos los fragmentos, IP coloca las porciones de cada fragmento en las posiciones relativas indicadas por los campos offset del encabezado de cada uno de ellos. El primer fragmento tendrá el campo "offset" puesto a 0, y el último fragmento tendrá la bandera de "más fragmentos" en 0.

Después que IP reconstruye el datagrama, la capa de red maneja a este paquete de información como si la red nunca lo hubiera fragmentado.

Ejemplos de Datagramas IP:

1.- Datagrama Simple

Este es el ejemplo del mínimo de datos que puede acarrear un datagrama de Internet



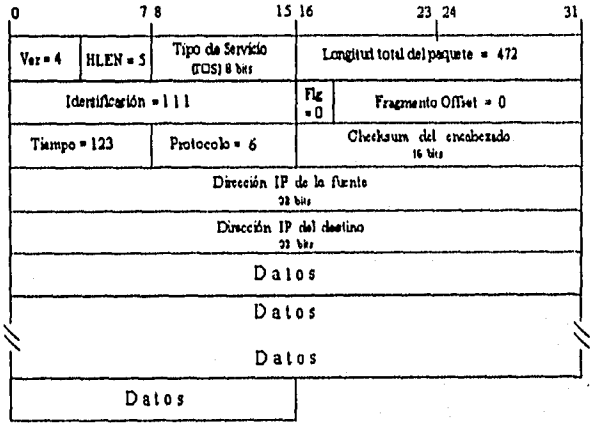
Datagrama Simple

2.- Datagrama que será fragmentado

En este ejemplo, primero se presenta un datagrama de tamaño moderado (452 bytes) de datos y después dos fragmentos que resultan de la fragmentación de dicho datagrama en caso de que el máximo tamaño permitido para la transmisión fuera de 280 bytes.

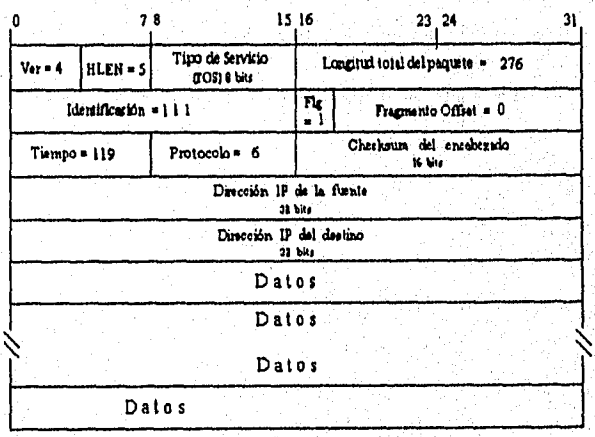
Como vemos en este ejemplo, el datagrama contiene 20 bytes de encabezado + un área de datos de 452 bytes = 472 bytes en total, y para poder transmitirse deberá ser fragmentado. De él se desprenderán dos fragmentos en múltiplos de 8 bytes como resultado de su fragmentación.

datagrama original:



Datagrama de Internet que será fragmentado

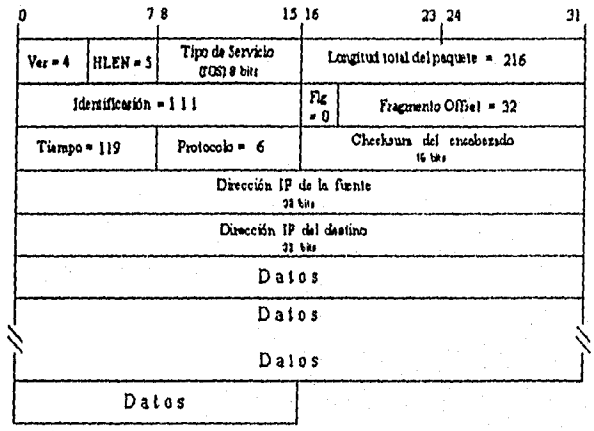
El primer fragmento (en múltiplos de 8 bytes) que resulta del rompimiento del datagrama original es de una longitud de 256 bytes de datos. Por lo tanto, la longitud total del paquete sería de 256 bytes de datos + 20 bytes del encabezado = 276 bytes.



Fragmento no. 1

La bandera de más fragmentos "Flg=1" indica que se esperan más fragmentos y el campo "Fragmento Offset=0" señala que los datos de este fragmento pertenecen al byte 0 (primer byte) dentro del datagrama original.

El segundo fragmento quedaría así:



Fragmento no. 2

En el fragmento 2 restan 196 bytes de datos y agregándole 20 bytes del encabezado, la longitud total del paquete sería de 216 bytes. El campo de "Fragmento Offset= 32" indica (en unidades de 8 bytes) el punto en el que se encuentran los datos de este fragmento dentro del datagrama original. En este caso, los datos deben colocarse en el byte 256 dentro del datagrama.

La bandera de más fragmentos "Flg=0" indica que este es el último fragmento de la serie.

2.2.2.- PROTOCOLOS DE TRANSPORTE

Generalmente para comunicarse con Internet, las aplicaciones necesitan intercambiar la información con la capa de transporte de TCP/IP. Esta capa de transporte incluye 2 protocolos: El Protocolo de Control de Transporte (TCP) y el Protocolo de Datagramas de Usuario (UDP).La fig. 2.6 señala el nivel donde se encuentran dichos protocolos:

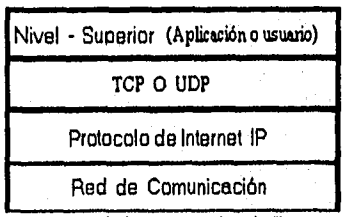


fig. 2.6 Nivel de los Protocolos de Transporte

Aplicaciones de Internet como el programa ftp, que transfiere archivos a través de la red, normalmente utilizan a TCP porque ofrece un servicio "fiable" de flujo de bytes. De esta forma, aplicaciones como e-mail también recurren a TCP por la misma razón. Aplicaciones con requisitos muy simples, tales como aquellos basados en el Protocolo de Transferencia de Archivos Trivial (TFTP), utilizan UDP.

Hemos visto que el módulo de IP entrega información entre computadoras hosts. Ahora analizaremos como la capa y los protocolos de transporte entregan información entre las aplicaciones. El Protocolo de Control de Transporte es un protocolo orientado a la conexión que utiliza un flujo "fiable" de bytes para transmitir y recibir información. Por su parte, el protocolo UDP es un protocolo no orientado a la conexión y "no fiable" que utiliza datagramas para enviar y recibir información.

La capa de transporte enruta paquetes desde y hacia programas de aplicación. De esta forma, la capa de transporte requiere de una forma para identificar cada aplicación, y es ahí donde surgen los "números de puerto". Cada aplicación, sin importar si son aplicaciones de servidor o de cliente, corresponden a un número de puerto único. Cuando un programa establece una sesión (conexión con la red), el programa es asignado a un número de puerto. En Internet existen algunos puertos asignados a aplicaciones o funciones específicas de la red y son denominados "puertos bien establecidos". Un puerto bien establecido es un puerto para un protocolo comúnmente usado por una aplicación o función de Internet. La siguiente tabla muestra los puertos de los protocolos más comunes en Internet:

Protocolo	*# Puerto*	*Protocolo*	*# Puerto*
Protocolo de Eco	7	Protocolo de Tiempo	37
Protocolo de Tiempo del día	13	Protocolo Whois	43
Protocolo de Transferencia de Archivos (FTP)	21	Protocolo de Transferencia de Archivos Trivial (TFTP)	69
Protocolo Telnet	23	Protocolo Finger	79
Protocolo de Transferencia de Correo Simple (SMTP)	25		

Puertos usados para Protocolos en Internet

Un puerto es como una dirección IP excepto que TCP/IP asocia a un puerto con un protocolo en vez de una computadora host.

Encapsulamiento

Kris Jamsa y Ken Cope mencionan en el libro Internet Programming, que *"el proceso de mover la información a través de la pila de protocolos es realmente un proceso de encapsulamiento de datos"*. La encapsulación simplemente requiere dar un formato a los datos para adaptarlos a un protocolo en particular. Mediante los datos fluyen por la pila de protocolos, cada capa o nivel de la pila construye su formato sobre la encapsulación de la capa previa. La figura 2.7 nos da un "vistazo" sobre el proceso entero del flujo de la información a través de la pila de protocolos TCP/IP.

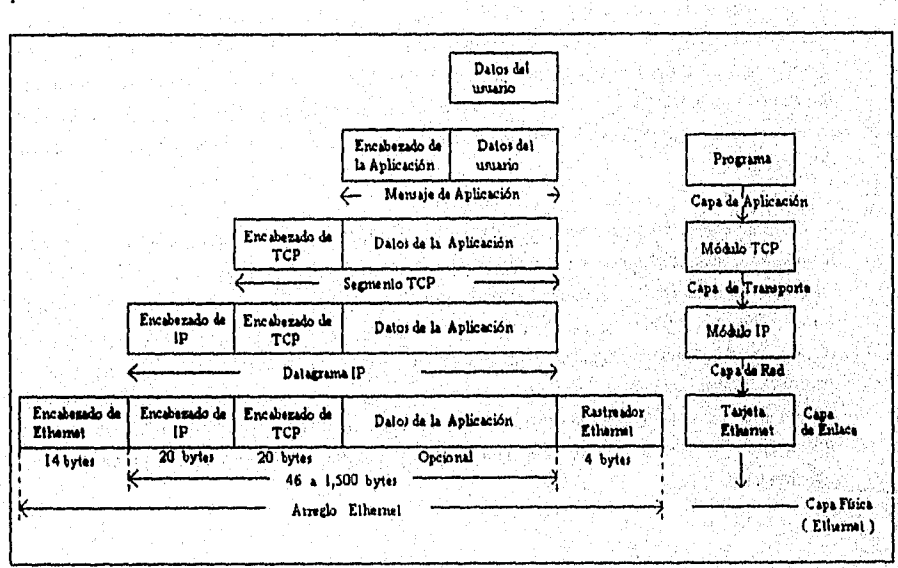


fig. 2.7 Encapsulamiento de los datos conforme fluyen a través de la pila de protocolos TCP/IP

Cuando el programador de Internet diseña sus programas lo hace como con cualquier otra aplicación. Cuando necesita transmitir información a través de Internet, deberá encapsular sus datos dentro del protocolo de transporte requerido por sus procedimientos. Para seleccionar el protocolo correcto sólo deberá saber que protocolos están disponibles y entender su funcionamiento.

PROTOCOLO PARA DATAGRAMAS DE USUARIO (UDP)

UDP es my similar a IP en que ambos son protocolos "no fiables" y sin conexión que utilizan datagramas en la entrega de información. IP entrega datos a un host, sin embargo, solamente UDP puede enrutar datos a múltiples destinatarios (programas de red) en un mismo host. Normalmente, la red asocia a esos destinatarios con un puerto de protocolo.

Un protocolo "no fiable" no asegura la entrega de los datos, es decir, el protocolo tratará de realizar la entrega, pero esto no garantiza que se lleve a cabo con éxito. Lo más importante es que un protocolo de transporte "no fiable" nunca notificará a la aplicación transmisora cuando el proceso de entrega ha fallado.

De otra forma, un protocolo que no es orientado a la conexión, no requiere establecer una conexión con otra aplicación antes de transmitir el mensaje. Como resultado, cada mensaje que utiliza un protocolo sin conexión debe contener toda la información para su entrega. Cada uno de estos mensajes contiene una dirección completa y exacta para realizar su entrega. Un protocolo sin conexión pasa el mensaje a la siguiente capa en la pila de protocolos y depende de la red para la entrega. UDP y el IP son protocolos "sin conexión".

Haciendo una comparación con el sistema de entrega postal (correo). Podemos pensar en IP como un camión de correo y los protocolos de transporte como los conductores de ese camión o carteros. Aunque el camión (IP) lleva los paquetes a sus direcciones correctas, en realidad es el cartero quién sortea y coloca las cartas en sus respectivos buzones.

Los camiones de correo (IP) entregan paquetes (datos) entre oficinas de correo (computadoras host). En la oficina de correo, los repartidores del correo o carteros (UDP) sortean el correo por no. postal de buzón (puertos). Después de sortear el correo, estas personas (UDP) colocan las cartas (datos) en los buzones correctos (puertos). Las personas dueñas de esos buzones checan periódicamente por su correo. Los carteros (UDP) no notifican a sus destinatarios (protocolos de aplicación) que estos tienen correo (datos); ellos simplemente depositan el correo en el buzón determinado (puerto).

Análogamente, el módulo UDP acepta datagramas que llegan y después los sortea y distribuye (demultiplexa) de acuerdo a los números de puerto de su destino. Esto queda ilustrado por la siguiente figura:

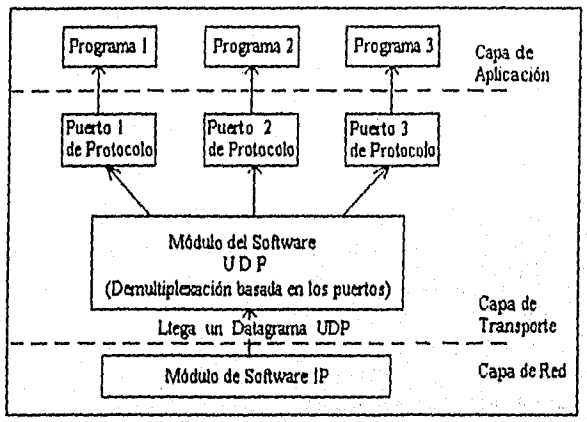


fig. 2.8 Flujo de datos a través del módulo UDP

UDP proporciona un procedimiento para que programas de aplicación puedan enviar mensajes a otros programas con un mecanismo mínimo de protocolo. El protocolo será orientado a la transacción de información; y la entrega y protección contra duplicación no están garantizados. Si las aplicaciones solicitantes necesitan una entrega fiable y ordenada de las corrientes de datos transmitidos, entonces deberán recurrir al protocolo TCP.

UDP también encapsula los datos recibidos y les agrega un encabezado propio del protocolo creando datagramas UDP. La estructura del encabezado de UDP es mucho más simple que la de datagramas IP. La fig. 2.9 se refiere al formato del encabezado UDP.

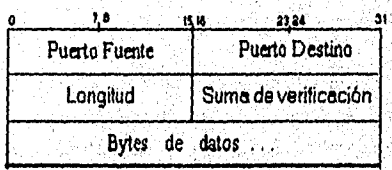


fig. 2.9 Formato del encabezado del datagrama UDP

El encabezado UDP utiliza únicamente 8 bytes de longitud (2 palabras de 32 bits).

CAMPOS

El puerto de la fuente es un campo opcional, cuando existe, este indica el puerto del proceso transmisor, y puede establecerse como el puerto al cual se debe direccionar una respuesta en la ausencia de otra información. Si no se encuentra, se inserta un valor de cero. El puerto de destino identifica la aplicación que recibirá los datos.

Longitud se refiere a la longitud en bytes de ese datagrama incluyendo el encabezado y los datos (esto indica que el mínimo valor de longitud es de ocho bytes).

Suma de Verificación del Encabezado
(Checksum)

Es el complemento en 16 bits del complemento de la suma de un seudo encabezado de información (proveniente del encabezado de IP), más el encabezado de UDP, y los datos, rellenando al final (si es necesario) con bytes en cero para hacer múltiplos de 2 bytes.

Una vez que se realiza el checksum, el seudo encabezado es desechado, pues sus parámetros formaran parte del encabezado IP.

El seudo encabezado conceptualmente antepuesto al encabezado UDP contiene la dirección de la fuente, la dirección del destino, el protocolo, y la longitud de UDP. Esta información previene la desviación de datagramas. Este procedimiento de chequeo es el mismo usado en TCP.

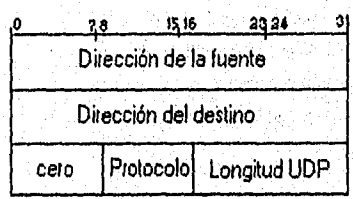


fig. 2.10 Seudo encabezado

El Checksum (suma de verificación) de UDP, a diferencia del producido por el encabezado IP, si incluye la información del datagrama en su elaboración de esta suma de verificación. El área de los datos sigue inmediatamente al encabezado UDP.

A pesar de que el checksum de UDP si incluye a los datos de su datagrama, el protocolo no requiere que sea calculado e incluido un chequeo de este tipo en su encabezado. Esto es al contrario de los protocolos IP y TCP los cuales sí requieren un checksum en su respectivo encabezado.

Interfaz para usuario

Una interfaz para el usuario debe habilitar la creación de nuevos puertos de admisión, recibir operaciones en los puertos receptores que producen los bytes de datos y una indicación del puerto y dirección de la fuente para ser enviados.

Interfaz IP

El módulo de UDP debe ser capaz de determinar la dirección de la fuente y el destino, así como el campo de protocolo desde el encabezado de IP. Una interfaz posible UDP/IP regresará el datagrama completo incluyendo todo lo del encabezado de Internet en respuesta a una operación recibida. Esta interfaz también habilita a UDP para transmitir un datagrama completo con el encabezado para que IP lo envíe. El protocolo IP verificará la consistencia de ciertos campos para computar la suma de verificación del encabezado (checksum).

Aplicación y número de Protocolo

La mayoría de los usos de este protocolo es para programas que envían mensajes muy cortos como el servidor de nombres de Internet y la transferencia de archivos trivial.

Este protocolo es el 17 (21 en octal) cuando se usa el protocolo IP.

PROTOCOLO DE CONTROL DE TRANSPORTE (TCP)

El Protocolo de Control de Transporte es usado con mayor frecuencia dentro de toda la serie de los protocolos TCP/IP. Así como el protocolo UDP, TCP transporta datos entre las capas de red y la de aplicación. Sin embargo, TCP es mucho más complejo que UDP y esto se debe a que provee un servicio de entrega orientado a la conexión mediante una corriente de bytes. Esto significa que TCP asegura que la entrega se realice y que la aplicación de destino reciba la información en el orden y secuencia adecuados.

Un protocolo orientado a la conexión debe establecer una conexión con otra aplicación antes de que ocurra cualquier comunicación. Este tipo de protocolos no puede comunicar o transportar datos hasta que establezca debidamente una conexión. TCP además de ser un protocolo orientado a la conexión, también es un protocolo "fiable".

Los protocolos "fiables" sí garantizan la entrega de los datos. Para asegurar la entrega, el protocolo intercambia mensajes de reconocimiento (ACK) entre las aplicaciones que efectúan la comunicación. Esto indica que, cada vez que un programa envía un mensaje, el programa espera a recibir un aviso donde se indique que el destinatario ya recibió la última información transmitida. Si el programa transmisor no recibe dicho mensaje de reconocimiento, el programa automática y repetidamente volverá a enviar el mensaje hasta que obtenga una respuesta positiva.

comunicación se completada, esta conexión es terminada y quiere decir que libera los recursos para otras

Protocolos TCP / IP

Un servicio de corriente-de-bytes transmite toda la información como una serie de bytes. Es decir, el protocolo trata a la información como una simple corriente de bytes sin importar la longitud de la información ni el número de transmisiones requeridas para enviar o recibir todos los datos. Estos protocolos también garantizan que el otro extremo de la conexión recibirá los datos en el mismo orden que la secuencia de transmisión.

Además, TCP es un protocolo que trata de optimizar el ancho de banda de la red. En otras palabras, TCP trata de maximizar la transmisión de datos a través de Internet. Para optimizar la transmisión de la red, TCP controla dinámicamente el flujo de datos entre las conexiones. De esta forma, si el buffer de datos del extremo receptor de la conexión comienza a saturarse, TCP indicará al extremo transmisor que reduzca su velocidad de transmisión.

Sabemos que el propósito principal de TCP es proveer una conexión segura y fiable entre pares de procesos y para brindar este servicio se requiere un buen manejo de :

- Transferencia de datos básicos
- Confiabilidad
- Control de flujo
- Conmutación
- Conexiones
- Precedencia y Seguridad

Transferencia de datos

TCP transfiere datos entre sus usuarios en un flujo constante de bytes, empaquetando cierto número de esos bytes dentro de segmentos para luego transmitirlos a través del sistema de Internet. En este modo TCP decide cuando obstruir o mandar la información a su conveniencia. TCP facilita al usuario el uso de registros, llamados cartas, para la transmisión. Cuando el usuario que envía la información indica un límite de registro (fin-de-carta), esto origina que TCP envíe y entregue rápidamente los datos hasta el destinatario.

Confiabilidad

TCP debe recuperar los datos que hayan sido dañados, perdidos, duplicados, o entregados fuera de orden por el sistema de comunicación de Internet. Esto se logra asignando una numeración secuencial a cada byte transmitido, y requiriendo un reconocimiento positivo (ACK-acknowledgment) del TCP receptor. Si el ACK no es recibido en un cierto intervalo de tiempo, los datos son retransmitidos. Ya en el receptor, la numeración secuencial facilita el ordenamiento de segmentos que pueden ir llegando en desorden y eliminar duplicados. Cada segmento transmitido es checado por el destinatario para descartar aquellos que han sido dañados.
La figura 2.11 ejemplifica el uso de mensajes de reconocimiento (ACK):

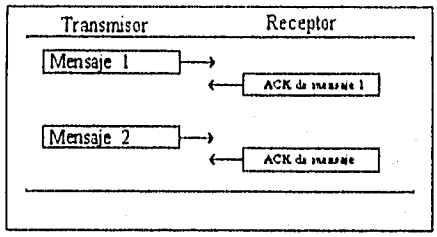


fig. 2.11 Transmisión de datos usando mensajes de reconocimiento ACK

Para incrementar la transmisión de mensajes, TCP no envía un mensaje y espera hasta recibir un reconocimiento antes de transmitir el siguiente. En cambio, TCP transmite varios mensajes antes de esperar la respuesta de sus reconocimientos.

Control de flujo

TCP deja que el receptor domine la cantidad de datos enviados por el transmisor. Esto se realiza mandando una "ventana" con cada ACK indicando un rango aceptable de número secuenciales más allá del último segmento recibido exitosamente. Para el modo de flujo, la ventana indica un número de bytes permitido para que el transmisor envíe antes de requerir otro permiso. Para el modo de registro, la ventana indica la cantidad permitida de espacio en el buffer que puede ocupar el transmisor, este puede ser mayor que el número de bytes transmitidos en caso de que haya una desigualdad entre el tamaño de la "carta" y el tamaño del buffer.

Conmutación

Para que varios procesos dentro de un mismo host puedan accesar simultáneamente las utilidades de TCP, este brinda un conjunto de direcciones o puertos (número que identifica una aplicación de Internet en particular) dentro de cada host. Los puertos eslabonados con las direcciones de la red y del host del sistema de comunicación de Internet conforman un "socket"(concatenación de una dirección de Internet con un puerto de TCP). Un par de sockets identifica cada conexión. Esto implica que cada socket puede ser usado simultáneamente en múltiples conexiones y la ligadura de puertos para procesos es manejada independientemente por cada host.

Conexiones

Los mecanismos de confiabilidad y de control de flujo requieren que los TCP's inicialicen y mantengan cierto estado de información para cada corriente de datos. La combinación de esta información incluyendo sockets, números secuenciales, y las ventanas, es llamada una conexión o enlace.

Cada conexión es únicamente especificada por un par de sockets identificando sus dos lados. Cuando dos procesos desean establecer comunicación, sus módulos TCP's deberán primeramente establecer una conexión (inicializar el estado de la información en cada lado), cuando la

comunicación es completada, esta conexión se termina o cierra para liberar los recursos para otros usos.

Debido a que las conexiones se establecen entre hosts no-fiables sobre el sistema de comunicación de Internet, se utiliza un mecanismo con números secuenciales basados cronológicamente (mediante reloj) para prevenir una inicialización errónea de conexiones.

Precedencia y Seguridad

Los usuarios de TCP pueden indicar la seguridad y precedencia de su comunicación. Sin embargo, se tienen valores preestablecidos para ser usados cuando no sea necesario modificarlos.

Hemos ilustrado como TCP para asegurar la fiabilidad y orden de la corriente de bytes, envía y recibe mensajes de reconocimiento. Pero, para efectuar estas operaciones, este protocolo debe tener un método para identificar la información transmitida. Es decir, la red debe sincronizar de alguna manera el extremo receptor con el extremo transmisor de la conexión TCP. Entonces ambos extremos de la conexión necesitan saber cuando pueden comenzar a transmitir datos y también como identificar la información del transmisor. Por ejemplo, supongamos que un módulo de TCP recibe un paquete de información alterado. Este módulo necesita señalarle al TCP transmisor cuál paquete debe volver a enviar. Para establecer una conexión, ambos extremos de la conexión deben negociar y acordar el uso de información de identificación que cada uno comprenda.

Los números secuenciales permiten a TCP que identifiquen la información en la corriente de bytes. Las computadoras-host utilizan una variedad de métodos para seleccionar el número secuencial inicial, y se puede pensar en ese número inicial como un número aleatorio.

El número secuencial inicial es simplemente un valor que uno de los extremos de la conexión transmite al otro. El extremo transmisor de la conexión le notifica al extremo receptor de que desea establecer una conexión y que va a iniciar la numeración (identificación) de su corriente de datos con "X" número. Cuando el otro extremo recibe la petición para establecer una conexión, entonces responde con un mensaje que incluye su propio número de secuencia inicial. TCP genera el número inicial del receptor completamente independientemente del número inicial del módulo transmisor.

Las conexiones TCP son conexiones "full-duplex", es decir, la información fluye en ambas direcciones al mismo tiempo. De esta forma, el flujo de la información en una dirección es independiente del flujo de información del otro extremo, y debido a esta capacidad de TCP, cada extremo de la conexión debe mantener 2 números de secuencia -uno para cada dirección del flujo de la información.

En el mensaje de respuesta inicial, el extremo receptor pone dos banderas en el encabezado del TCP: la bandera de Sincronización (SYN) para decirle al módulo TCP transmisor que anote el número secuencial del receptor. El módulo TCP que recibe también establece la bandera de Reconocimiento (ACK) para decirle al que envía que examine el campo de número de reconocimiento.

El módulo TCP destinatario utiliza el número secuencial recibido del otro extremo para crear un número de reconocimiento. Un número de reconocimiento siempre especifica el próximo número secuencial que la conexión espera recibir. Así, en su mensaje de respuesta inicial, el extremo que recibe almacena el número secuencial del transmisor más uno "1". Por ejemplo, si el módulo fuente que pide la conexión envía un número secuencial de 1,000; en respuesta, el módulo de destino almacenará el número 1,001 en el campo de reconocimiento para su primer mensaje de respuesta. En otras palabras, para el extremo que pide la conexión (cliente), el módulo TCP que la recibe (servidor) siempre le avisará sobre el próximo número que espera recibir (1,001).

Establecimiento de Conexión (OPEN)

Antes de que empiece a transferir información, el TCP del extremo "cliente" que requiere la conexión debe notificar del mensaje de respuesta del módulo TCP "servidor". De esta forma, cuando el TCP cliente recibe este mensaje de respuesta, también debe enviar un reconocimiento del reconocimiento. En realidad el módulo cliente está reconociendo la petición del módulo servidor para la sincronización.

El mensaje enviado por el módulo cliente también especificará la bandera de reconocimiento, y en su campo de número de reconocimiento este módulo almacenará el número secuencial enviado por el módulo servidor en su mensaje inicial de respuesta más uno. De esta manera se lleva a cabo lo que se conoce como "un triple apretón de manos" antes de que TCP establezca una conexión oficial:
1.- El TCP cliente solicita una conexión mandando una petición de sincronización y un número de secuencia inicial.'

2.- El módulo TCP servidor reconoce la petición de una conexión y, al mismo tiempo, solicita que el módulo cliente se sincronice con el número de secuencia inicial de el módulo del servidor.

3.- El extremo cliente reconoce la petición del servidor para realizar la sincronización.

Después de este triple apretón de manos, ambos lados de la conexión tienen toda la información que requieren para identificar la información en el canal de comunicación (números secuenciales y de reconocimiento). Esto significa que ambas partes han sincronizado sus números de secuencia y han notificado la sincronización, y ahora sí comenzará la transmisión de la información.

Cerrando una Conexión (CLOSE)

Los programas cierran conexiones de TCP usando un "doble apretón de manos". Cualquier extremo de la conexión puede iniciar el cierre de la conexión. Para cerrar la conexión, un extremo de la conexión envía un mensaje con la bandera de Termino o Final (FIN). Sin embargo, debido a que TCP opera con una comunicación full-duplex (datos en ambas direcciones), los programas deben terminar cada flujo de información en su dirección correspondiente y de forma independiente. Esta bandera de FIN determina que dicho extremo ha terminado de enviar información, pero aún puede recibir datos del otro extremo. El mensaje de reconocimiento del otro extremo de la conexión indica que ambos lados han acordado terminar el flujo de información en una dirección. Cerrar una conexión TCP es un proceso de 2 pasos: primero, uno de los extremos desarrolla un "cierre activo" y el otro lado efectúa un "cierre pasivo". El extremo que envía la bandera FIN es el que realiza el cierre activo y el extremo que recibe la bandera de termino, iniciará un cierre pasivo. Un cierre pasivo significa que este extremo de la conexión también envía un mensaje con la bandera FIN. Después de que los dos extremos de la conexión han mandado las banderas de FIN y recibieron su reconocimiento correspondiente, la conexión termina oficialmente.

Encabezado de TCP

Como podemos observar en la siguiente ilustración, el encabezado de TCP es mucho más complejo que el del protocolo UDP. A continuación describiremos los campos que constituyen el encabezado de TCP.

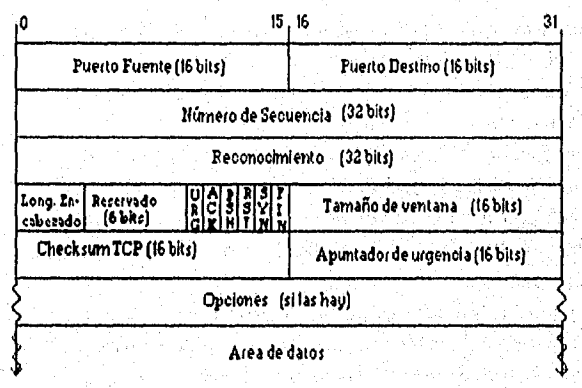


fig. 2.12.- Estructura del Segmento TCP

Puerto de la Fuente y del Destinatario

Estos campos de puerto son de 16 bits cada uno e identifican las aplicaciones transmisoras y receptoras. Los números de puerto más las direcciones IP (del encabezado IP) de ambos fuente y

receptor, se combinan para identificar cada conexión TCP como única. Podemos referirnos a cada extremo de una conexión de TCP como un socket.

Números de Secuencia

Este campo de 32 bits identifica el primer byte de información en el área de los datos del segmento TCP. TCP identifica a un byte por su "offset" relativo desde el inicio de la corriente de datos. Se puede identificar a cada byte en una corriente de datos a través de un número de secuencia.

Número de Reconocimiento

Este campo de 32 bits determina el próximo byte de los datos que la conexión espera recibir de la corriente de información. Por ejemplo, si el último byte recibido fue el número secuencial 1000, TCP mandará un número de reconocimiento de 1001.

Longitud del Encabezado

Así como el encabezado de IP, el campo de longitud del encabezado TCP utiliza 4 bits para especificar esa distancia en palabras de 32 bits. También el encabezado de TCP es generalmente de 20 bytes de longitud. El área de los datos comienza inmediatamente después del encabezado. Al examinar este campo, los módulos TCP receptores pueden calcular el inicio del área de los datos.

Banderas

El encabezado de TCP incluye seis campos de banderas de un-bit:

URG Esta bandera especifica al TCP receptor que los datos son urgentes y debe procesarlos antes que otros.

ACK Esta bandera determina un número válido de reconocimiento. Este campo ayuda a TCP para asegurar la fiabilidad de la información.

PSH Esta bandera (push) indica al módulo TCP que inmediatamente mande el segmento a su aplicación de destino.

RST Este campo (reset) solicita al módulo TCP que borre la conexión. Esto se hará cuando se detecte un problema con la conexión.

SYN Esta bandera permite que el TCP receptor se sincronice con los números de secuencia. Este campo dice al TCP receptor que el transmisor está preparado para transmitir una nueva corriente de datos.

FIN Esta bandera indica al receptor que el transmisor ha terminado de enviar información. Esta bandera solo termina el flujo de datos en la dirección que esta maneja. El TCP que la recibe debe transmitir un mensaje con la bandera FIN para poder cerrar la conexión completamente.

Tamaño de Ventana

Este campo de 16 bits especifica al módulo TCP receptor el número de bytes que el transmisor está dispuesto a aceptar. Este valor es variable de acuerdo al ancho de banda de la red.

TCP Checksum (verificación)

Este campo de 16 bits incluye los datos de TCP en su cálculo. TCP requiere que el transmisor calcule e incluya sus "checksums" en este campo. De la misma forma, TCP necesita que los módulos receptores verifiquen estos campos cuando reciben los datos.

Apuntador Urgente

Es un campo de 16 bits que determina la localización de un byte en el área de los datos. El propósito de la bandera URG y de este campo, es el de notificar al módulo TCP receptor que algún tipo de información urgente existe y señalar con el apuntador su localización.

Opciones

Los módulos de TCP generalmente usan el campo de Opciones con la opción de "Máximo Tamaño del Segmento". El tamaño máximo del segmento TCP es similar al de la capa física (MTU-máxima unidad de transferencia) ya que define el mensaje más largo que TCP puede aceptar. Si un módulo de TCP no transmite un número con el tamaño máximo de segmento, TCP asume el tamaño máximo por default de 536 bytes.

Ambiente del Host

TCP está diseñado para ser un módulo en un sistema operativo de tiempo compartido. Los usuarios accesan a TCP tanto como accesan al sistema de archivos. TCP puede llamar a otras funciones del sistema operativo, por ejemplo, al manejo de estructuras de datos.

El siguiente diagrama ilustra el lugar de TCP en la jerarquía de los Protocolos en Internet.

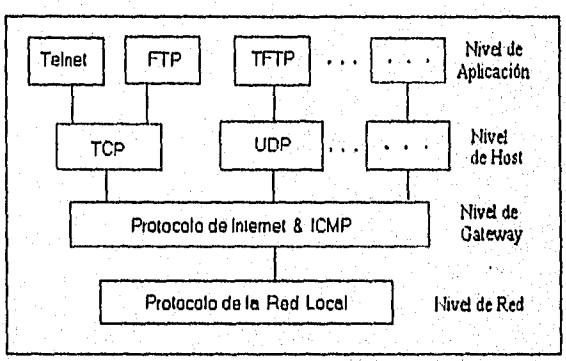


fig. 2.13 Lugar del Protocolo TCP en la pila de Protocolos TCP/IP

TCP tiene la capacidad de dar soporte a protocolos de niveles superiores y se supone fácil el comunicarlo con protocolos como ARPANET Telnet.

La interfaz actual de la red es supuesta a ser controlada por un módulo controlador de dispositivo. TCP no llama directamente al controlador de dispositivo de la red, en su lugar llama al módulo del protocolo de datagrama (IP) que en su turno realiza la petición al manejador de dispositivo.

Aún cuando se supone que los procesos son soportados por el sistema operativo del host, los mecanismos de TCP no excluyen utilerías de TCP para un procesador de comienzo-a-fin. Sin embargo, en tal operación, un protocolo de host de comienzo-final debe brindar la funcionalidad para dar soporte al tipo de interfaz TCP/usuario utilizada.

La interfaz TCP/usuario permite acceso a peticiones del usuario en el TCP para abrir o cerrar una conexión, para mandar o recibir datos, o para obtener el estado relativo de una conexión. Estas peticiones son como las que realizan los programas del usuario en un sistema operativo, por ejemplo, aquellas para abrir, leer, y cerrar un archivo.

La interfaz TCP/Internet provee peticiones para mandar y recibir datagramas direccionados a módulos de TCP en cualquier host del sistema de Internet. Estas llamadas contienen parámetros para el manejo de direcciones, tipo de servicio, precedencia, seguridad y otra información de control.

Protocolos TCP / IP

2.2.3.- PROTOCOLO DE MENSAJES DE CONTROL DE INTERNET (ICMP)

Hemos revisado cómo el protocolo IP es usado en el servicio de datagramas de host-a-host en un sistema de redes interconectadas. Los dispositivos que enlazan a estas redes son los gateways. Estos gateways se comunican entre sí, con el propósito de un buen control, a través del Protocolo Gateway a Gateway (GGP). En ocasiones un gateway o un host de destino se comunican con un host transmisor, por ejemplo, para reportar un error en el procesamiento de un datagrama. Para estos casos es necesario hacer uso del Protocolo de Mensajes de Control (ICMP). Este protocolo usa como soporte básico a IP como si fuera un protocolo de nivel superior, sin embargo, ICMP es en realidad una parte integral de IP, y debe implementarse en cada módulo de IP. De la misma forma que IP, ICMP es un protocolo "no fiable" y que no está orientado a la conexión.

Cualquier host TCP/IP puede emplear al Protocolo ICMP para transmitir mensajes de errores en la red, de control, y de información hacia otro host en la red. Las redes TCP/IP también encapsulan los mensajes ICMP en datagramas IP. Esto implica que los mensajes ICMP viajan a través de una red TCP/IP dentro del área de datos de un datagrama IP. Sin embargo, el destino de un mensaje ICMP siempre es un módulo de software de la capa de red, nunca una aplicación específica de usuario o de la red.

Los mensajes de ICMP son transmitidos en diversas situaciones: por ejemplo, cuando un datagrama no puede alcanzar su destino, cuando un gateway no tiene capacidad suficiente de buffers para transmitir un datagrama, cuando se puede dirigir al host para mandar el tráfico en una ruta más corta, cuando el campo de "tiempo de vida" de un encabezado IP ha llegado a 0, etc..

El protocolo IP no está diseñado para ser absolutamente fiable. El propósito de estos mensajes de control es el proporcionar retroalimentación (feedback) sobre problemas en el ambiente de comunicación, no para hacer a IP más fiable.

Aún así, no hay garantía de que un datagrama sea entregado o que un mensaje de control sea informado. Algunos datagramas todavía pueden ser perdidos sin ningún mensaje que lo reporte.

Los protocolos de nivel superior que usa IP deben implementar sus propios procedimientos de fiabilidad cuando se requiere de una comunicación segura.

Los mensajes de ICMP generalmente reportan errores en el procesamiento de datagramas. Para prevenir el regreso infinito de mensajes sobre los mensajes, ningún mensaje de ICMP es enviado sobre los mensajes de ICMP. Esto significa que no se generan mensajes de error ICMP para datagramas que contienen mensajes ICMP.

FORMATOS DE MENSAJE

TCP/IP encapsula cada mensaje ICMP dentro de un datagrama IP. El software de red identifica a cada mensaje ICMP por dos valores: un campo de tipo y un campo de código. Estos dos valores conforman los 2 primeros campos del encabezado ICMP.

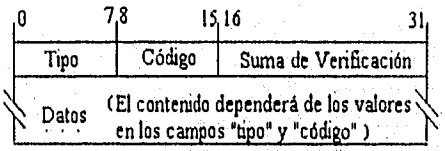


fig. 2.14 Encabezado ICMP

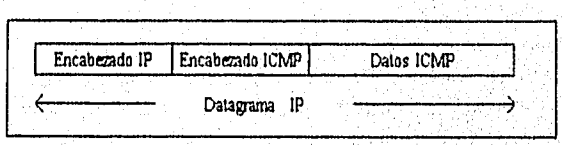


fig. 2.15 Encapsulamiento para ICMP

Los mensajes de ICMP son mandados usando el encabezado básico de IP. A menos que difiera a las descripciones del formato individual, los valores de los campos del encabezado de IP son los siguientes:

- *Versión* 4
- *HLEN* (Longitud del encabezado en palabras de 32 bits)
- *Tipo de Servicio* 0
- *Longitud total*

Longitud del encabezado de Internet y los datos en bytes

- *Identificación, banderas y fragmento offset*

Usados en la fragmentación

- *Tiempo de vida*

Tiempo de vida en segundos; debido a que este campo se decrementa en cada máquina donde el datagrama es procesado, el valor en este campo debe ser por lo menos del número de gateways que el datagrama cruzará.

- *Protocolo*

ICMP = 1

- *Suma de verificación del encabezado*

Es el complemento de 16 bits de la suma del complemento de todas las palabras de 16 bits en el encabezado. Para calcular esta suma de verificación, su campo deberá estar en cero. Esta verificación será reemplazada más tarde.

- *Dirección de la fuente*

La dirección del gateway o host que edita el mensaje de ICMP. A menos que se especifique lo contrario, esta puede ser cualquier dirección de un gateway.

- *Dirección del destino*

La dirección del gateway o host donde el mensaje será enviado.

El primer byte de la porción de datos del datagrama es un *campo de tipo* de ICMP; el valor de este campo determina el tipo de mensaje y el formato de los datos sobrantes. Cualquier campo etiquetado como "sin usar" está reservado para futuras extensiones y debe estar en cero cuando sea enviado, pero los receptores no deberán usar estos campos (excepto para incluirlos en la suma de verificación). Siguiendo a los dos primeros campos, ICMP incluye el campo de 16 bits para la suma de verificación. Como sabemos, estos campos ayudan a detectar la corrupción de los datos. En esta suma de verificación se incluye todo el mensaje ICMP, no solo su encabezado.

La información que prosigue a estos 3 campos dependerá del tipo de mensaje ICMP en particular. Los mensajes ICMP usados para reportar errores incluyen el encabezado IP y los primeros 64 bits de los datos del datagrama que ocasionó el error.

A continuación se presenta un sumario de los tipos de mensajes que pueden ser emitidos por ICMP :

SUMARIO DE TIPO DE MENSAJES

Tipo	*Descripción*
0	respuesta a un eco
3	destinatario inalcanzable
4	inhibición de la fuente
5	redirección
8	eco
11	tiempo excedido
12	problema de parámetros
13	marca de tiempo
14	respuesta de marca de tiempo
15	petición de información
16	respuesta de información
37	petición de nombre de dominio
38	respuesta de nombre de dominio

La siguiente sección muestra el formato de cada uno de estos mensajes.

MENSAJE DE DESTINO INALCANZABLE

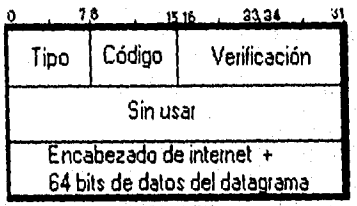


fig. 2.16

Campos de IP:

- Dirección de destino.- La dirección de la red transmisora del datagrama original.

Campos de ICMP:

- Tipo.-

El no. 3 denota al mensaje de destino inalcanzable

- Código.- Puede ser una de las causas siguientes:

0 = red inalcanzable

1 = host inalcanzable

2 = protocolo inalcanzable

3 = puerto inalcanzable

4 = necesidad de fragmentación, pero la bandera de "no fragmentar" está en 1

5 = fracaso en el enrutamiento de la fuente

- Suma de verificación

Esta verificación es el complemento de 16 bits de la suma del complemento del mensaje empezando con el campo "tipo" del ICMP. Para calcular esta verificación, este campo debe estar en cero. Más tarde será reemplazado.

- Encabezado + 64 bits de datos de datagrama

El encabezado IP más los primeros 64 bits de los datos del datagrama donde ocurrió el error. Estos datos son usados por el host para colocar el mensaje en el proceso apropiado. Si un protocolo de nivel superior utiliza números de puertos, se asume que estos se encontrarán en los primeros 64 bits del datagrama.

- Descripción

Si de acuerdo con la información en las tablas de enrutamiento de los gateways, la red especificó en el campo de destino de un datagrama que el destinatario es inalcanzable, por ejemplo, la distancia de la red es infinita, el gateway debe enviar un mensaje de destino inalcanzable al host fuente del datagrama. Además, en algunas redes, el gateway puede determinar si el host de destino es inalcanzable. Los gateways de estas redes envían mensajes de destino inalcanzable cuando el host de destino no puede ser alcanzado.

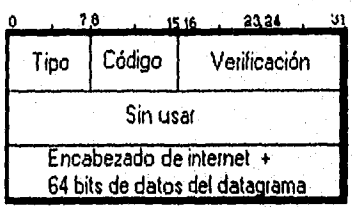
Si en el host de destino el módulo de IP no pudo entregar el datagrama porque el módulo del protocolo indicado en el puerto del proceso no está activo, el host de destino debe mandar un mensaje de destino inalcanzable al host fuente.

Otro caso es cuando un datagrama debe ser fragmentado para su transmisión en un gateway aunque exista la bandera de "no-fragmentar". En esta caso el gateway debe descartar el datagrama y regresar un mensaje de destino inalcanzado.

Los códigos 0, 1, 4 y 5 deben transmitirse desde un gateway.

Los códigos 2 y 3 son transmitidos por un host.

MENSAJE DE TIEMPO EXCEDIDO



(fig. 2.16)

Campos de IP:

- Dirección de destino.- La red fuente y dirección de los datos del datagrama original.

Campos de ICMP:

- Tipo

11 El no. 11 es el identificador del mensaje de tiempo excedido

- Código.- Las causas pueden ser:

0 = Tiempo de vida excedido en la transmisión

1 = Tiempo de ensamblaje de fragmentos excedido

- Verificación

Igual que en otros mensajes

- Encabezado de internet + 64 bits de datos de datagrama
- Descripción

Como se analizó al principio de este capítulo, el encabezado IP de un datagrama incluye un campo "Tiempo de Vida" (TTL) que limita el tiempo de existencia de un paquete en la red. Cada gateway que recibe y retransmite un datagrama decrementa el valor del campo TTL.

Si el gateway que procesa al datagrama encuentra que el campo "tiempo de vida" es cero, entonces deberá descartar al datagrama, y tendrá que notificar al host fuente con un mensaje de tiempo excedido (tipo 11).

Si un host que está ensamblando un datagrama fragmentado no puede completar la reconstrucción dentro del límite de tiempo, entonces descarta el datagrama y manda un mensaje de tiempo excedido.

El código cero debe ser emitido por un gateway y el código 1 por un host.

MENSAJE DE PROBLEMAS DE PARÁMETROS

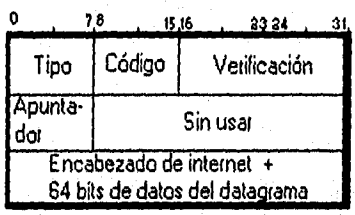


fig. 2.17

Campos :

de IP ● Dirección de destino

ICMP ● Tipo 12 Este tipo de mensaje cae en el no. 12

● Código 0 Mal Encabezado IP

1 Ausencia de una opción requerida

● Verificación

● Apuntador

Si el código = 0 , identifica el byte donde se detectó el error.

● Encabezado de internet + 64 bits de datagrama

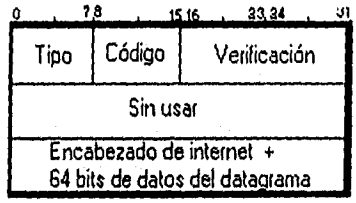
● Descripción

Si el gateway o host que están procesando a un datagrama encuentran un problema con los parámetros del encabezado y no pueden continuar con el proceso, el datagrama será descartado. Una causa potencial de este problema es con los argumentos incorrectos en una opción. El gateway o host deben notificar al host de origen mediante un mensaje de problemas con parámetros. Este mensaje sólo se envía si el error causó que el datagrama fuera descartado.

El apuntador identifica el byte del encabezado original del datagrama donde fue detectado el error (puede localizarse en la mitad de una opción). Por ejemplo, 1 indica que algo está mal con el "tipo de servicio" y (si existe la presencia de opciones) 20 indica que algo anda mal con el código de tipo de la primer opción.

El código 0 será enviado por un gateway o un host.

MENSAJE DE FUENTE INHIBIDA (QUENCH)



(fig. 2.16)

Campos:

IP • Dirección del destino

ICMP • Tipo 4

• Código 0 El código siempre será cero

• Verificación

• Encabezado de internet + 64 bits de datagrama

• Descripción

Los gateways trabajan con datagramas IP que llegan a través de la red y estos paquetes pueden saturar la capacidad del dispositivo para su control.

Una congestión en el tráfico de la red ocurre cuando un gateway no puede manejar todos los paquetes que llegan . Cuando esto sucede, los gateways comienzan a descartar paquetes que van llegando y envían un mensaje de "fuente inhibida" (tipo 4) hacia los hosts transmisores. Esto indica al transmisor que existe una congestión en el tráfico de la red y que debe reducir la velocidad de transmisión.

El gateway puede enviar un mensaje de este tipo por cada datagrama que descarte. En recibo de un mensaje de fuente inhibida, el host de origen debe recortar la velocidad a la cual está enviando tráfico al destino especificado hasta que ya no reciba más mensajes desde el gateway. El host de origen puede ir incrementando gradualmente la velocidad de transmisión hasta que nuevamente reciba mensajes de fuente inhibida.

El gateway o el host pueden transmitir el mensaje cuando se aproximen al límite de capacidad en lugar de esperar hasta que la capacidad sea excedida. Esto significa que el datagrama que activó el mensaje de fuente inhibida sí debe ser entregado.

El código 0 será emitido por un gateway o un host.

MENSAJE DE REDIRECCIÓN

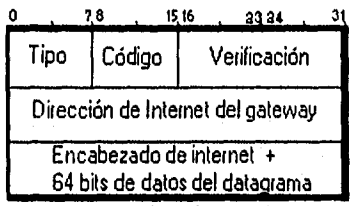


fig. 2.18

Campos:

IP • Dirección del destino

ICMP • Tipo 5

• Código

0 = redireccionar datagramas a la red

1 = redireccionar datagramas al host

2 = redireccionar datagramas a el "tipo de servicio" y la red

3 = redireccionar datagramas a el "tipo de servicio" y el host

• Verificación

• Dirección de internet del gateway

Dirección del gateway del cual será enviado el tráfico para la red especificada en el campo de la red de destino de los datos del datagrama original. Indica al host receptor qué gateway usar en el futuro.

•Encabezado de internet + 64 bits de datagrama

•Descripción

Este mensaje surge de un gateway para informarle al host transmisor que existe una ruta más directa para enviar un datagrama a su destino. Un gateway, G1, recibe un datagrama desde un host en una red en la cual el gateway está conectado. El gateway G1 checa su tabla de enrutamiento y obtiene las direcciones del próximo gateway , G2, en la ruta a la red de destino del datagrama ,X. Si G2 y el host de destino están en la misma red, un mensaje de redirección es mandado al host fuente. El mensaje de redirección informa al host que envíe su tráfico a la red X directamente por el gateway G2 al ser un ruta más corta para el destino. El gateway transmite el datagrama original a su red de destino.

Para datagramas con las opciones de "enrutamiento de la fuente" de IP y la dirección del gateway en el campo de la dirección del destino, no es enviado un mensaje de redirección aún cuando haya una ruta mejor para el destinatario final que la próxima dirección en la ruta de la fuente.

Los códigos 0, 1, 2 y 3 serán enviados desde un gateway.

MENSAJE DE ECO O DE RESPUESTA DE ECO

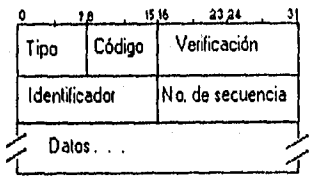


fig. 2.19

Campos:

IP •Direcciones

La dirección de una fuente en un mensaje de eco será el destino del mensaje de la respuesta al eco. Para formar un mensaje de respuesta de eco, las direcciones de la fuente y el destino son invertidas, el código de tipo se pone a cero, y la verificación es recalculada.

ICMP •Tipo 8 para el mensaje de eco

0 para el mensaje de respuesta al eco

•Código 0

•Verificación

Si la longitud total es impar, los datos recibidos son rellenados con un byte de ceros para computar la suma de verificación. Esta verificación será reemplazada en el futuro.

•Identificador

Si el código es 0, un identificador que auxilie en la unión de ecos y respuestas deberá ser cero.

•Número de secuencia

Si el código es 0, un número secuencial para auxiliar en la unión de ecos y respuestas será cero.

•Descripción

ICMP tiene algunas funciones que permiten a los administradores de una red el detectar problemas. Los mensajes de ICMP para petición y respuesta de eco son un ejemplo de ello. Cuando el módulo ICMP de un host recibe una petición de eco, este transmite un mensaje de respuesta de eco que es idéntico al mensaje de petición. El mensaje de respuesta indica al transmisor que el host que recibió la petición de eco está activo y respondiendo a mensajes de la red.

Los datos recibidos en el mensaje del eco deben ser regresados en el mensaje de respuesta de eco. El identificador y el número de secuencia pueden ser usados por el transmisor para ayudar en la unión de las respuestas con las peticiones de eco. Por ejemplo, el identificador puede ser usado como un puerto en TCP ó en UDP para identificar una sesión, y el número de secuencia se incrementará en cada solicitud de eco transmitida. El transmisor del eco regresa los mismos valores en cada respuesta de eco.

El código 0 debe ser enviado desde un gateway o un host.

MENSAJE DE MARCA DE TIEMPO O
RESPUESTA A MARCA DE TIEMPO

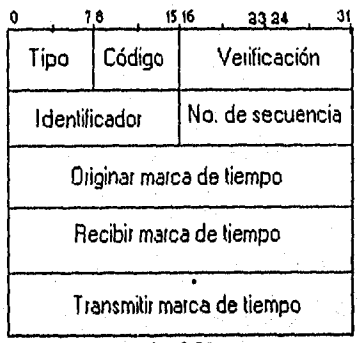


fig. 2.20

Campos:

IP •Direcciones

La dirección de la fuente en un mensaje de marca de tiempo será el destino del mensaje de respuesta de una marca de tiempo. Para formar este mensaje de respuesta, las direcciones de la fuente y el destino son invertidas, el código de tipo cambia a 14 y la suma de verificación es recalculada.

ICMP •Tipo 13 para el mensaje de marca de tiempo
14 para la respuesta de marca de tiempo

•Código 0

•Verificación

•Identificador

Si el código es 0, un identificador de apoyo para la unión de marcas de tiempo y sus respuestas, será cero.

•Número de secuencia

Si el código es 0, un número de secuencia para unir marcas de tiempo y respuestas, será cero.

•Descripción

Este tipo de mensaje permite a los profesionales de la red estimar el tiempo de tránsito de un paquete entre hosts.

El transmisor usa los campos "identificador" y "no. de secuencia" para distinguir entre múltiples peticiones de marcas de tiempo. Los tres campos de marca de tiempo representan el no. de segundos pasada la medianoche. Antes de que transmita el datagrama, el host transmisor llena el campo de "originar marca de tiempo" con la hora en ese momento. El receptor llena el campo "recibe marca de tiempo" tan pronto como el datagrama llega. Adicionalmente, el receptor llena el campo de "transmitir marca de tiempo" antes de transmitir su respuesta.

La marca de tiempo originada es el tiempo en que el transmisor tocó por última vez el mensaje antes de mandarlo, la marca de tiempo que se recibe es el tiempo en que el eco lo tocó por primera vez en el recibo, y la marca de tiempo en la transmisión es el tiempo en que el eco tocó por última vez el mensaje cuando era enviado.

El identificador y el número de secuencia puede ser usado por el transmisor del eco como apoyo en la unión de las respuestas y las peticiones. Por ejemplo, el identificador puede usarse como un puerto en TCP o UDP para identificar una sesión, y el número de secuencia es incrementado en cada petición enviada. El destinatario regresa estos mismos valores en la respuesta.

El código 0 puede ser recibido desde un gateway o un host.

MENSAJE DE PETICIÓN DE INFORMACIÓN
O RESPUESTA DE INFORMACIÓN.

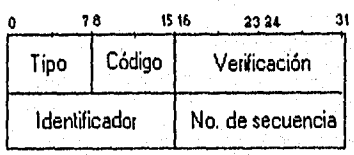


fig. 2.21

Campos:

IP •Direcciones

La dirección de una fuente en un mensaje de petición de información será el destino del mensaje de respuesta de la información. Para formar un mensaje de respuesta, las direcciones de la fuente y el destino son invertidas, el código de tipo es cambiado a 16, y la suma de verificación es calculada nuevamente.

ICMP •Tipo 15 mensaje de petición de información
16 mensaje de respuesta de información

•Código 0

•Verificación

•Identificador

Si el código es 0, un identificador auxiliar en el enlace de respuestas y peticiones es 0.

•Número de secuencia

Si el código es 0, un número de secuencia para el enlace de peticiones y respuestas es 0.

•Descripción

Este mensaje debe enviarse con la red fuente en el encabezado de la fuente y los campos cero de la dirección del destino.

El módulo de IP que responde debe enviar la respuesta con las direcciones completamente especificadas. Este mensaje es una forma para encontrar el número de red en la que se encuentra.

El identificador y el número de secuencia son usados por el transmisor del eco para facilitar el enlace de las respuestas con las peticiones. Por ejemplo, el identificador puede ser usado como un puerto en TCP o UDP para identificar una sesión, y el número de secuencia se incrementa en cada petición enviada. El destinatario devuelve los mismos valores en la respuesta.

El código 0 puede recibirse desde un gateway o un host.

MENSAJES DE NOMBRES DE DOMINIO

Cada nombre de dominio es expresado como una secuencia de etiquetas. Cada etiqueta es representada como un campo de longitud de un byte, seguido por el número de bytes. Debido a que cada nombre de servidor termina con la etiqueta nula del root, un nombre de servidor es finalizado por un byte de longitud de cero. Los 2 bits más significativos de cada byte de longitud debe ser '00' y los 6 bits restantes limitan con 63 bytes o menos.

Cuando los 2 bits más significativos del byte de la longitud son '11', la longitud se interpreta como una secuencia de 2 bytes , indicando un 'offset' (compensación) desde el comienzo del mensaje (campo de tipo).

Para simplificar las implementaciones, la longitud total del nombre del dominio (incluyendo bytes de etiqueta y bytes de longitud de etiqueta) es restringida a 255 bytes o menos.

PETICIÓN DE NOMBRE DE DOMINIO

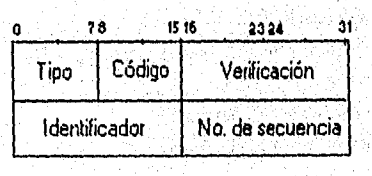


fig. 2.22

- Tipo 37
- Código 0
- Verificación

La verificación de ICMP

- Identificador

Si el código es 0, un valor para enlazar peticiones y respuestas será 0.

- Número de secuencia

Si el código es 0 el valor será 0

•Descripción

Una petición de nombre de dominio es usada separadamente por cada destinatario IP consultado (queried).

Una petición de nombre de dominio de ICMP recibida con un destinatario broadcast o multicast debe descartarse discretamente.

En recibo a un mensaje de error de ICMP, las implementaciones atentarán resolver el nombre del dominio usando el método IN-ADDR.

RESPUESTA AL NOMBRE DEL DOMINIO

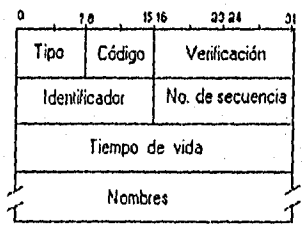


fig. 2.23

•Tipo 38

•Código 0

•Verificación

Verificación del ICMP

•Identificador

Copiado desde la petición

•Número de secuencia

Copiado desde la petición

•Tiempo de vida

El número de segundos que el nombre será guardado.

•Nombres

Cero o más nombres de dominios completamente calificados. La longitud de este campo es determinado por la longitud total del datagrama.

Cuando ningún nombre es reconocido, el campo se elimina (longitud cero), pero la respuesta es enviada como indicación de autorización de que ningún nombre sea conocido. Cuando más de un nombre es conocido, todos esos nombres serán enlistados.

Cualquier nombre que no ajuste completamente dentro de la respuesta MTU no es enviado.

El campo de la fuente IP en una respuesta debe ser la misma que la del destinatario correspondiente del mensaje de petición.

Cada host y enrutador deben implementar una función de servidor de nombres de dominio que reciba peticiones de estos nombres y envíe las correspondientes respuestas a los mismos.

Un host también debe implementar una aplicación que sea interfaz para enviar peticiones de nombres de dominio y reciba respuestas de los nombres, esto para propósitos de diagnóstico.

CONSIDERACIONES DE SEGURIDAD

El propósito fundamental de esta especificación es proveer un mecanismo en cuanto a direcciones para la resolución de nombres. Este mecanismo está sujeto al uso de los protocolos de seguridad de IP por la autenticidad y privacía.

Aunque la infraestructura de enrutamiento al destinatario no provee seguridad dentro y fuera de sí, es al menos tan fiable como la entrega de correspondencia para otras sesiones con el mismo punto (peer).

Una señal criptográfica del DNS, localizada para el uso de la respuesta en la transmisión de la dirección DNS, puede usarse para verificar la misma respuesta.

Capítulo III
Dominios y Servidores

3.1.- DIRECCIONES DE INTERNET

Una dirección de Internet es generalmente conocida como una dirección IP. La mayoría de las veces se asocia a estas direcciones con las computadoras host. Sin embargo, hay que aclarar que en realidad no es la computadora en esencia la que tiene una dirección dentro de Internet, sino la tarjeta interfaz de la red. En otras palabras, la dirección corresponde a la tarjeta interfaz (por ejemplo aquellas de tecnología Ethernet) que conecta a la computadora con la red, y requieren tener una dirección única dentro de la red . Esto significa que cada interfaz dentro de Internet debe corresponder con una dirección IP única.

Algunas ocasiones un host puede contener más de una tarjeta interfaz y esto implica que un solo host de Internet puede tener asociadas varias direcciones IP (una por cada tarjeta). Pero, debe quedar claro que una tarjeta interfaz sólo tendrá una dirección y esta deberá ser única dentro de todo el sistema de Internet, es decir que ninguna dirección IP podrá ser duplicada.

Una dirección de Internet es de una longitud de 4 bytes (32 bits) que frecuentemente son escritas en notaciones decimales separando cada byte mediante un punto. Estas direcciones también aparecen en su notación binaria o hexadecimal.

El siguiente ejemplo representa las diversas notaciones de una misma dirección IP:

En número binario	10000100 11111000 00101100 01111000
En número hexadecimal	0x84F82C78
En notación decimal (con punto)	132.248.44.120

Nota: De forma similar al lenguaje de programación C, se agregará el prefijo 0x para distinguir valores hexadecimales.

La siguiente tabla ayudara a entender la relación entre cada byte de la dirección IP:

Decimal	Hex	Binario
132	0x84	10000100
248	0xF8	11111000
44	0x2C	00101100
120	0x78	01111000

Podemos observar que en cualquier caso los valores son equivalentes, aunque por facilidad la notación decimal es más utilizada en la representación numérica de las direcciones IP.

Una dirección IP consta de 32 bits los cuales determinan el número de la red y el número del host (tarjeta interfaz) conectado a la misma. Para distinguir una red de otra dentro de las miles pertenecientes a todo Internet, existe un Centro de Información de Red de Internet (InterNIC) que se encarga de que cada red tenga un número identificador de red único. En el diseño de una dirección IP, el byte de orden más significativo (el primero en el lado izquierdo) representa el número de la red y los bytes restantes identifican el número del host.

Las direcciones IP son leídas de izquierda a derecha, donde el comienzo de esta dirección señala a que red forma parte y el extremo derecho determina a que computadora o host de esa red pertenece.

Un campo de dirección que contiene todos los bits en 1's (255) representa una dirección de "broadcast" (un mensaje destinado a todas las computadoras de la red). Un campo con todo en 0's (000) representa una dirección que determina a "esta" red y "este" host. Internet reserva estas 2 direcciones para el uso exclusivo de estos campos.

Debido a que una dirección típica de Internet es escrita en cuatro grupos de números (4 bytes), sus valores de cada grupo no exceden de 255, ya que es el valor máximo que puede adoptar un byte (11111111). Sabiendo que el grupo de orden superior es el que identifica el número de una red, esto supone que solo se podrían interconectar 255 redes con 16,777,216 hosts cada una. Para vencer esta limitación de espacio de direcciones, los profesionales de Internet idearon un arreglo simple pero muy efectivo de manejar estas direcciones. Para simplificar la administración de estas direcciones, el Centro de Información de la Red las ha dividido en clases. Estas clases especifican cuantos bytes de la dirección serán usados para determinar el número ID (identificador) de la red. La siguiente tabla nos muestra como se clasifican estas clases:

Clase	Bits de orden Superior	Bytes disponibles para el ID de la Red
A	0----	1 byte
B	10---	2 bytes
C	110--	3 bytes
D	1110	(usado para multicasting)
E	1111	(reservado para futuro uso)

Tabla 3.1 Clases de direcciones IP

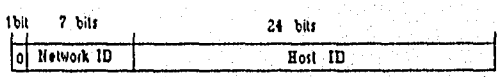
Una red requiere que cada tarjeta interfaz que pertenezca físicamente a ella, tenga el mismo número ID de red, pero un único número ID de host. La tabla siguiente apunta el tamaño de la red en cada clase:

Clase	Red en un rango de:	Bits para el número de Red	N bits sobrantes para subred y host
A	0 - 127	8 bits	24 bits
B	128 - 191	16 bits	16 bits
C	192 - 223	24 bits	8 bits
D	224 - 239	*Multi cast*	
E	240 - 255	*Reser vado*	

Tabla 3.2 Números de Red y Tamaño

CLASE "A"

La primer clase de una dirección, clase "A", tiene un máximo de un byte para determinar el tipo de clase (0) y el número de red. Esto deja 3 bytes (24 bits) para identificar el número del host.

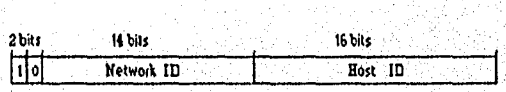


DIRECCION CLASE "A"

En la tabla 3.1 de las clases de direcciones IP se puede observar que la clase A utiliza un bit de orden superior para codificar la clase. Esto libera solo 7 bits (01111111) disponibles para los números ID de red, lo que implica que sólo se puedan manejar 127 redes[9] (2^7-1) dentro de esta clase . Sin embargo, como las redes de esta clase usan un espacio de 24 bits para la dirección del host, cada red de este tipo puede conectar teóricamente 16,777,216 (2^{24}) hosts. De ahí que las redes que necesitan conectar más de 65,536 (2^{16}) hosts utilizan la clase "A".

CLASE "B"

Como se muestra en la tabla 3.1, las direcciones de Clase B utilizan un máximo de 2 bytes para representar su tipo de clase y el número ID de la red, y dejan un espacio de 16 bits para los números ID de host.



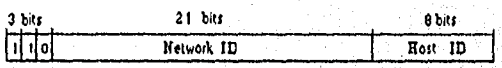
DIRECCION CLASE "B"

[9] *No son 128 redes pues recordemos que la dirección cero está reservada*

Después de que se sustraen los 2 bits de orden superior usados para codificar el tipo de clase (10), se tienen 14 bits disponibles para identificar los números ID de red. De esta forma, Internet puede conectar 16,384 (2^{14}) redes dentro de las direcciones de la Clase "B". Utilizando 16 bits para el número identificador de host, cada red dentro de esta clase puede conectar en teoría hasta 65,536 (2^{16}) hosts. Las redes que necesitan conectar más de los 65,536 hosts requieren de la Clase "A". El Centro InterNIC reserva las direcciones de la Clase "B" para redes que esperan conectar por lo menos 256 (2^{8}) computadoras hosts.

CLASE "C"

Este tipo de direcciones emplean como máximo 3 bytes para especificar la clase y el número ID de red, lo que indica que se tienen sólo 8 bits para poner el número ID del host.

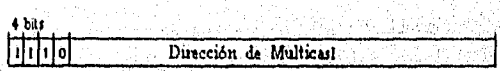


DIRECCION CLASE "C"

Restando los 3 bits de orden superior usados para codificar la clase de dirección, se tienen 21 bits libres para denotar los números ID de red. Esto permite que Internet pueda conectar un número de 2,097,152 (2^{21}) de redes individuales que están bajo la Clase "C". Sin embargo, debido a que estas direcciones solo tienen 8 bits libres para los números de host, Internet limita a cada una de estas redes a contar con menos de 256 (2^{8}) computadoras. En pocas palabras, esto significa que redes pequeñas son las que utilizan direcciones de Clase "C".

CLASE "D" Y "E"

El Centro InterNIC reserva las direcciones de Clase "D" para direcciones de multicast. Una dirección de multicast representa un grupo de computadoras hosts dentro de Internet, donde la palabra "multicasting" determina que un mensaje será entregado a uno o más hosts de la red.



DIRECCION CLASE "D"

Ninguna dirección es permitida con los 4 bits de orden superior 1111. Estas direcciones son de la clase "E". El InterNIC reserva las direcciones de la Clase "E" para usos futuros.

Si realizamos un cálculo total de todos los números ID disponibles para red, encontraremos que teóricamente Internet es capaz de enlazar más de 2 millones de redes individuales:

127 (A) + 16,384 (B) + 2,097,152 (C) = 2,113,663 redes

Si cada computadora host contiene solamente una tarjeta interfaz de red y cada red conectara el máximo número de computadoras, entonces Internet sería capaz de trabajar con más de 3 mil 741 millones de computadoras.

127 (16,777,216) + 16,384 (65,536) + 2,097,152 (256) = 3,741,319,168 hosts

Y aunque ahora estas cifras pueden sonar exageradas, muchos de los diseñadores de redes se encuentran desarrollando nuevas propuestas para extender el número de direcciones de Internet en un espacio aún mayor.

Haciendo una comparación del espacio de direcciones de Internet con y sin la codificación de clases, observaremos que con el esquema original de las direcciones, usando sólo un byte (2^8) para identificar el número de red y 3 bytes (2^{24}) para el número de host, el sistema de Internet podía conectar arriba de 4 mil millones de computadoras.

*(255 * 16,777,216) = 4,278,190,080 computadoras*

No obstante, todas esas computadoras únicamente podrían ser parte de sólo 255 redes. Empleando el esquema de las direcciones basadas en clases, Internet reduce en un 10% el número de hosts dispuestos, pero por otro lado, incrementa el número de 255 redes a más de 2 millones. De esta forma Internet sacrifica unas cuantas direcciones de hosts para acrecentar tremendamente el número de identificadores de red.

Comparación de direcciones IP	No. de redes	No. de hosts
Con Codificación	2,113,663	3,741,319,168
Sin Codificación	255	4,278,190,080

Direcciones Especiales

Existen direcciones especiales que cumplen con funciones diferentes a las de identificar a un host individual:

La dirección de red=0 se interpreta como "este", dentro de "esta" red. Es decir, la dirección *0.0.0.27* podría identificarse como el host 27 dentro de "esta" red.

La dirección host=0 ó host=255 son interpretadas como "todos" (señal de broadcast). Esto significa que la dirección *132.248.255.255* ó *132.248* se interpretan como "todos" los hosts de la red *132.248*.

La dirección de red=127 de la clase "A" es asignada como función de "loopback" que identifica a la misma red y casi siempre se denota como la dirección *127.0.0.1*. Cuando un datagrama enviado por una aplicación o protocolo de nivel superior va dirigido a la misma red, se coloca la dirección *127.0.0.1* que indica que el datagrama deberá viajar en el mismo host.

3.1.1.- ASIGNACIÓN DE DIRECCIONES IP

Como ya hemos señalado anteriormente, el número posible de direcciones IP es verdaderamente extenso. De tal forma que si cada dirección IP debe ser única en todo el sistema de Internet, debe haber alguien responsable de realizar esta tarea y asegurar la validez de cada una de las 3 mil 700 millones de direcciones posibles. Para llevar a cabo la administración de todos estos aspectos relativos a las direcciones IP, existe el Centro de Información de la Red de Internet "InterNIC", el cual se encarga de asignar todos los números ID de red y asegura su existencia como única. Dentro de cada red, el administrador de la misma tendrá la tarea de asignar los números identificadores de los hosts.

El encargado de cada red tiene la responsabilidad de administrar la disposición y monitoreo de sus elementos, así como de asignar las direcciones de los hosts dentro de la red. En México existen varias instituciones conectadas a Internet y cada una de ellas cuenta con un Centro de Información de la Red para asignar y asegurar sus direcciones individuales. En RedUNAM, por ejemplo, existen los servicios de DNS y NOC/NIC

DNS (Domain Name Service)
El Servicio de Nombres de Dominio es un sistema cuya tarea es la de convertir las direcciones lógicas de IP a sus respectivos nombres de máquinas y viceversa, de tal forma que el usuario pueda utilizar nombres dentro de la red en lugar de direcciones (números).Por ejemplo, **condor.dgsca** en vez de **132.248.10.3**.

Su función básica es la de proporcionar información sobre los dispositivos o máquinas conectadas a la red en respuesta a una solicitud de un cliente.

NOC/NIC

El Centro de Operación de la Red y Centro de Información de la Red son servicios que llevan a cabo el monitoreo y mantenimiento de la red, así como las asesorías requeridas por los usuarios de la misma.

El Centro de Operación de la red (NOC) determina si el uso de la red cae dentro de las normas establecidas por esa institución, en caso contrario, se notificará al administrador responsable de la dependencia o red externa para que se tome una acción y se remedie inmediatamente. Este centro tiene como tarea principal la de brindar el Servicio de Nombres de Dominio o DNS para todo el campo Universitario e Instituciones conectadas a ella con cobertura nacional. Y habilita la conexión de estas instituciones al sistema de Internet.

El Centro de Información de la Red (NIC) tiene como propósito el de proveer información administrativa y de soporte, primordialmente a los usuarios de su red, y secundariamente a los usuarios del sistema de Internet y de otras agencias de servicio conectadas directamente.

Cada NIC Internet debe:

a) Proporcionar fuentes de información.- Las fuentes de información son los diferentes elementos (discos, cintas, etc.) que se utilizan para almacenar información disponible a los usuarios, tales como boletines, archivos en línea, etc.

b) Soporte a usuarios finales.- El propósito primordial de un NIC es prestar información de la red a sus usuarios finales. Para llevarlo a cabo se puede emplear: soporte telefónico, correo electrónico, transferencia electrónica de información, etc. Todo con el objeto de contestar dudas y solucionar ciertos problemas dentro del área provista.

c) Información de la red

d) Mantenimiento de soporte a la infraestructura NIC.- Cada NIC del sistema debe dar soporte de la infraestructura de NIC/INTERNET. Debe participar en las sesiones del IETF del Grupo de Trabajo para Servicios de Usuarios (USWG) para discutir y encontrar soluciones a procesos de servicio de red. Y debe formar parte del nic-forum, una lista de correo electrónico donde cada NIC ofrece soluciones y envía información de interés a través del correo.

3.1.2.- DIRECCIONES DE SUBRED

Después de que el Centro InterNIC asigna el número ID de redes, los administradores de cada red son quiénes deben asignar los números ID de sus hosts. En esta tarea los administradores cuentan con cierta flexibilidad cuando configuran sus redes. Mientras se cumpla que cada tarjeta interfaz cuente con una sola y única dirección IP, ellos pueden usar el espacio de las direcciones de sus hosts de la manera que mejor les convenga. Los administradores de la red pueden subdividir el espacio de sus direcciones de host para crear una red local de redes (subred). Por ejemplo, si un administrador tiene a su cargo una red de Internet que funciona bajo la Clase "B", entonces contará con 16 bits disponibles para los números ID de host. De este espacio, puede subdividir esos 16 bits en 2 bytes, utilizando un byte como identificador de la red local (subred) y el otro byte como identificador del host. De esta forma, el administrador de la red puede crear lo que se conoce como una subred.

La dirección de una subred es cualquier dirección derivada de un esquema de subredes, y es sólo válida dentro de la red donde ellas forman parte.

RedUNAM, por ejemplo, maneja direcciones IP de la clase "B", su número identificador de red es *132.248*. La red de la UNAM está constituida por varias redes de facultades, institutos y demás dependencias que la conforman. Para asignar una dirección distintiva a cada una de estas redes, entonces se crean las direcciones de subred. Las direcciones de las subredes permiten identificar a cada dependencia por un número. Por ejemplo:

La Dirección General de Servicios de Cómputo Académico (DGSCA) maneja direcciones como las siguientes:

132.248.10.1

132.248.10.3

La Escuela Nacional de Estudios Profesionales Aragón identifica a sus máquinas con direcciones del tipo:

132.248.44.120

132.248.44.160

Sabemos que los dos primeros grupos de números de una dirección de clase "B" identifican el número de la red y para todos los ejemplos anteriores este identificador es el mismo (*132.248.*). Los dos grupos restantes forman parte del identificador de host, pero al usar direcciones de subredes, el tercer grupo denota el número de subred y el grupo final determina el número de host.

Las direcciones *132.248.10.1* y *132.248.10.3* corresponden a máquinas ubicadas en las instalaciones de DGSCA en la UNAM. El número identificador de esta dependencia es el *10* (número de subred).

Las direcciones *132.248.44.120* y *132.248.44.160* pertenecen a servidores conectados a RedUNAM, pero ubicados físicamente en la ENEP Aragón. Esto indica que pertenecen a la subred local de la ENEP Aragón que a su vez forma parte de RedUNAM.

Para los cuatro casos la dirección de la red es el número *132.248* que identifica a RedUNAM, el número de subred *10* pertenece a DGSCA y el número *44* es el identificador de la subred Aragón, y los números restantes señalan el número ID de cada host.

Teóricamente, el administrador de la red podrá crear una subred de 254 redes interconectadas, cada una con 254 hosts (no son 256 porque los valores que contienen todos 1's y todos 0's están reservados). Frecuentemente, los administradores de red utilizan las direcciones de subredes para permitir a una dirección específica de Internet extenderse a más de una red física. Los sistemas que son enlazados a otras redes envían paquetes a las direcciones de Internet. Dentro de las subredes, los enrutadores internos emplearán esas direcciones de subred para enrutar los datos a la dirección física correcta. En otras palabras, las redes manejan direcciones de subred de forma interna, estas subredes son visibles dentro de su red, pero invisibles desde redes externas.

3.1.3.- CATEGORÍAS DE DIRECCIONES IP

Las direcciones IP caen dentro de 3 categorías: *unicast, broadcast* y *multicast*. Se puede hacer referencia a las Clases A, B y C como clases de **unicast**, ya que identifican a un simple host en particular. Las direcciones 0 y 255 (todos 0's o todos 1's) son reservadas para **broadcast**. Una dirección de broadcast especifica que los elementos conmutadores de paquetes (switches) enrutarán la información a todos los hosts de la red, es decir, broadcast implica que la entrega de los mensajes será efectuada en todas las computadoras de una red en particular.

La dirección 254 se utiliza para especificar la dirección IP del enrutador (gateway) por default. Esta convención implica que si una dirección de un host es : *132.248.44.120*, su enrutador o gateway por default tendrá una dirección IP de: *132.248.44.254* .

Una dirección de **multicast** identifica un grupo específico de hosts en el sistema de Internet. Este grupo de hosts se puede extender a múltiples redes y puede incluir un número ilimitado de computadoras. Este grupo de hosts es dinámico porque una computadora host puede unirse y dejar un grupo de hosts como este requiera. Para aplicaciones tales como conferencias interactivas, se puede hacer uso del multicasting pues el usuario podrá enviar información a múltiples computadoras, pero no necesariamente todas dentro de la misma red. Los hosts y los enrutadores que soportan la acción del multicasting emplean el Protocolo de Manejo de Grupos de Internet (IGMP).

3.2.- PROTOCOLOS DE LAS DIRECCIONES IP

En la figura 2.2 del capítulo anterior pudimos observar que la capa de enlace incluye dos protocolos de direcciones: el Protocolo de Resolución de Direcciones (ARP) y el Protocolo de Resolución de Direcciones Inverso (RARP).

En la capa física se manejan las direcciones de la tarjeta interfaz (Ethernet) las cuales son de 6 bytes de ancho en comparación con los 4 bytes de las direcciones IP. Toda la información transmitida a través de una red que usa tecnología Ethernet debe hacer uso de arreglos de información de Ethernet. Las tarjetas interfaz no prestan atención en las direcciones IP, sino que buscan esos arreglos en la red para determinar su propia dirección Ethernet[10].

Los protocolos de la serie TCP/IP sólo trabajan con las direcciones IP, mientras que la tarjeta interfaz que usa arreglos de Ethernet sólo trabaja con direcciones de Ethernet. La diferencia entre este tipo de direcciones representa un problema de comunicación de la red. Y para ello se encuentran los protocolos ARP y RARP que solucionan este problema resolviendo (traduciendo) las direcciones. Estos protocolos traducen una dirección IP a su dirección de la capa de enlace y viceversa. El diagrama siguiente ilustra la función de cada protocolo.

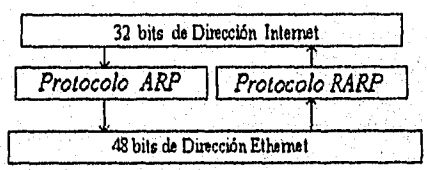


fig. 3.1 Protocolos de Resolución de Direcciones que convierten direcciones IP y de capa de enlace

Protocolo de Resolución de Direcciones (ARP)

El módulo del Protocolo de Resolución de Direcciones crea un mapa (tabla) y relaciona las direcciones en la capa de red (direcciones IP) con su correspondiente dirección en la capa de enlace. La dirección de la capa de enlace es específica del tipo de tecnología empleada. Por ejemplo, las direcciones de Ethernet son de 6 bytes de ancho, las direcciones de IBM Token Ring son de 2 ó 6 bytes y las direcciones ARCnet son de sólo un byte.

---

[10] Toda Tarjeta tiene una dirección única asignada desde su fabricación (dirección física). Ethernet tiene direcciones de 6 bytes escritos en hexadecimal y separados por dos puntos (:).

Las configuraciones de las redes pueden cambiar de manera en que los hosts se conectan o dejan la red. Afortunadamente, el protocolo ARP realiza los mapas de las direcciones en forma dinámica, es decir, que ARP automáticamente re-mapea las direcciones cuando la configuración de la red cambia. En términos generales, ARP utiliza la capacidad de broadcasting de la capa de enlace para examinar la red e identificar las computadoras que dejan o se adhieren a la red.

Protocolo de Resolución de Direcciones Inverso (RARP)

Como su nombre lo indica, este protocolo relaciona o traduce direcciones de la capa de enlace tales como direcciones de Ethernet a sus direcciones IP. Esta conversión depende de la tecnología de red empleada (Ethernet, Token Ring, ARCnet, etc.)

RARP fue diseñado para ser usado por computadoras que no tienen unidades de disco, es decir estaciones de trabajo que pueden leer su dirección de capa de enlace desde su tarjeta interfaz de red.

Al hacer uso del protocolo RARP, una estación de trabajo puede mandar una petición broadcast que solicite a otro host de la red que averigue la dirección de la capa de enlace y le reporte la dirección IP correcta de dicha estación de trabajo sin unidad de disco. Por medio de su dirección IP, la estación de trabajo puede mandar un mensaje broadcast pidiendo a otro sistema o servidor que cargue el sistema operativo de la estación. De esta forma es que se pueden enlazar estaciones de trabajo sin unidad de disco al sistema de Internet y después cargar el sistema desde lugares remotos a través de la red.

3.3.- SISTEMA DE NOMBRES DE DOMINIO (DNS)

Una definición de dominio es el territorio sobre el cual se ejerce control y se establecen reglas. En comunicaciones, se refiere a todos los recursos que están bajo el control de un sistema de computación. En un sistema con jerarquías, dominio denota a un grupo denominado con un cierto nombre y que tiene control sobre los grupos que se encuentran debajo de él.

Generalmente el hecho de referirse a una máquina por medio de direcciones IP se considera manejable entre gente de cómputo. Sin embargo, para los usuarios comunes de Internet, estas direcciones resultan una serie de números tediosa que en ocasiones llegan a olvidar. La mayoría de la gente encuentra más fácil usar y recordar nombres que una serie de números. Por esta razón fue creado el Sistema de Nombres de Dominio (DNS), el cual permite que los usuarios de Internet puedan referirse a computadoras hosts por medio de nombres en lugar de una dirección.

Al igual que las direcciones IP, los nombres de las computadoras deben ser únicos dentro de todo el Sistema, es decir, debe cuidarse que no se dupliquen. Aunque los nombres funcionan bien entre los humanos, en realidad las computadoras necesitan forzosamente conocer las direcciones IP para poder realizar la transmisión de información. Para esto debe de haber una forma de traducir los nombres de las máquinas a sus respectivas direcciones IP (función que corresponde al Sistema DNS).

En el comienzo de Internet la asignación de los nombres era fácil a través de un registro que manejaba el NIC de la red (Network Information Center) . Estos nombres eran dados de alta a través de una forma que era enviada electrónicamente (por correo), y el NIC se hacia cargo de mantener y actualizar el archivo de nombres y direcciones. Este archivo llamado "hosts", que contenía los nombres de las máquinas con sus respectivas direcciones IP, era distribuido regularmente a cada servidor de la red y los nombres eran palabras simples que cada administrador elegía para ser únicos. Si se utilizaba un nombre, la computadora buscaba en el archivo "hosts" y lo substituía por su dirección correspondiente.

Cuando Internet empezó a multiplicar su tamaño, también lo hizo el tamaño del archivo "hosts". Esto representaba significativos retrasos en encontrar un nombre registrado y por lo tanto también dificultaba controlar que los nombres no se duplicaran. Además era mucho tiempo el que se requería para distribuir este grandísimo archivo a cada máquina que debía actualizarlo. Obviamente surgió la necesidad de crear un sistema distribuido que facilitara el control de este archivo y la velocidad con que iba cambiando. Esta base de datos distribuida es el DNS.

Una base de datos distribuida físicamente graba datos en 2 o más computadoras del sistema. Para los programas que utilizan esta información, la ubicación geográfica de las computadoras es irrelevante. Es decir que, la base de datos puede incluir archivos grabados en computadoras localizadas en California y Nueva York. El software de la base de datos maneja y controla la colección completa de estos datos como una sola base de datos. De esta manera es como funciona el Sistema de Nombres de Dominio de Internet (DNS) bajo el modelo de cliente/servidor.

El DNS (Domain Name System) es un sistema que proporciona un método para distinguir la asignación y mantenimiento de nombres mediante grupos jerárquicos llamados dominios. Un nombre estará formado por el nombre de la máquina y los dominios a los que pertenece. Cada elemento dentro de este nombre es conocido como una etiqueta o dominio. Por ejemplo, el nombre de la máquina *condor.dgsca.unam.mx* consta de 4 etiquetas: **condor**, **dgsca**, **unam**, y **mx**. Las etiquetas de los nombres en Internet son separados con un punto que se lee como la palabra "punto". Otro ejemplo es el nombre *ftp.microsoft.com* que consiste de 3 etiquetas: **ftp**, **microsoft**, y **com**.

Este método facilita la administración de los nombres asignándole a los grupos (etiquetas) diferentes grados de responsabilidad con los subsecuentes elementos del nombre. Cada nivel en este sistema es conocido como dominio. Por ejemplo, considerando el nombre *ftp.microsoft.com*, la etiqueta **ftp** indica que la computadora es un elemento que soporta las operaciones de transferencia de

archivos (ftp). La etiqueta **microsoft** describe la organización o entidad a la cual pertenece esa computadora (Microsoft Corporation). Y por último la etiqueta **com** determina que esa organización hace uso de la computadora para fines comerciales. Así, podemos observar que cada etiqueta de un nombre de computadora es un dominio que describe una esfera de actividad, asunto, o función.

Puede haber un número variable de dominios dentro de un nombre, pero prácticamente no rebasan de 5 etiquetas.

Mientras que las direcciones IP de una máquina (siempre de 4 bytes de longitud) tienen un orden desde el nivel más alto hasta el inferior (de red a host), los nombres IP están estructurados primero desde la parte inferior hasta el nivel superior (de la máquina a la red). Veamos el ejemplo de la estructura típica de los nombres IP dentro de una universidad :

- *host.departamento.institución.edu*

  ejemplo: *ux.cso.uiuc.edu*

  ux es el nombre de una máquina con una dirección IP única, el nombre de esa máquina ha sido creado y lo mantiene el grupo **cso** que es el departamento donde se ubica dicha computadora. El departamento cso es parte de la Universidad de Illinois (**uiuc**) que a su vez pertenece al dominio educativo (**edu**) en los Estados Unidos.

- *host.campus.institución.lugar*

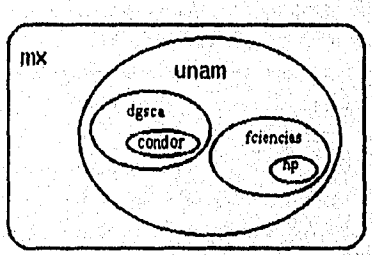
  ejemplo 1: *hp.fciencias.unam.mx*

  hp es una computadora con dirección IP *132.248.28.3* ubicada físicamente en la Facultad de Ciencias (**fciencias**) de la UNAM (**unam**) en México (**mx**).

  ejemplo 2: *condor.dgsca.unam.mx*

  condor es un host perteneciente a la Dirección General de Servicios de Cómputo Académico (**dgsca**) de la UNAM (**unam**) en México (**mx**).

Las direcciones IP de RedUNAM se encuentran bajo el dominio EDU el cual se divide por países y luego por instituciones académicas, siguiendo por último el nombre de la máquina o host. Veamos el siguiente diagrama:



Autoridad de Dominio

3.3.1.- ESTRUCTURA DEL DNS DE INTERNET

El Servicio de Nombres de Dominio introduce el concepto de zonas al modelo de red de Internet. Una zona es una comunidad jerárquica de hosts, administrados por una autoridad particular y servidos por un conjunto de servidores de nombres. Esta comunidad puede incluir hosts individuales, más cada uno de los servidores de nombres y sus clientes (subzona) localizados debajo del grupo de nombres de servidores autorizados para esa zona. Las zonas generalmente representan fronteras o límites administrativos, tales como el dominio administrativo de una red local.

La estructura del DNS de Internet puede compararse al organigrama de una organización. En la punta del organigrama se encuentra un punto de comienzo conocido como nivel de "root" (la raíz), después sigue un nivel secundario donde cada elemento es un dominio que tiene un nombre. A su vez, cada dominio puede subdividirse en sub-dominios, etc. A continuación se presenta la estructura jerárquica del Sistema de Dominio de Nombres de Internet.

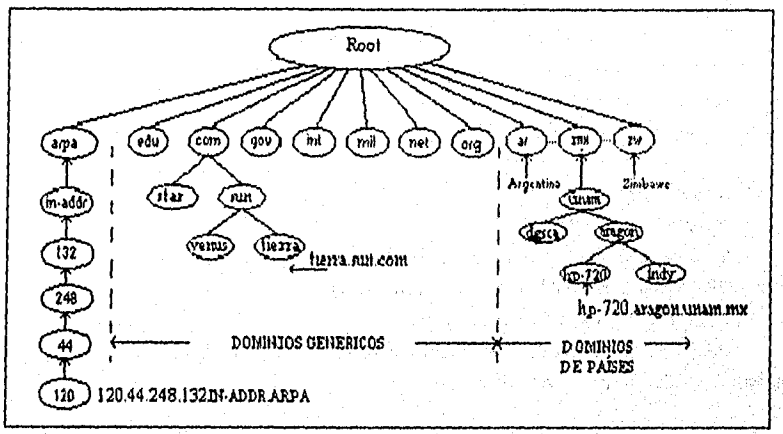


fig. 3.2 Estructura jerárquica del DNS de Internet

El nivel secundario a la raíz del DNS lo encabezan 3 dominios superiores :

1.- **Arpa** que es un dominio de Internet que transforma las direcciones IP (en notación decimal con punto) a sus nombres de dominio, en lugar que los nombres de dominio se conviertan a sus direcciones IP decimales con punto. Este dominio es conocido como IN-ADDR.ARPA y contiene esencialmente la misma información como la del espacio de un nombre de host, pero este es expresado en términos de direcciones IP.
Un nombre en el dominio IN-ADDR.ARPA tiene cuatro etiquetas precediéndole, y que corresponden a los 4 bytes de una dirección IP. Esta dirección es listada de derecha a izquierda. Por ejemplo, un host cuya dirección IP sea *132.248.44.120* tendrá un nombre de dominio IN-ADDR.ARPA:
120.44.248.132.IN-ADDR.ARPA

2.- El grupo **genérico u organizacional** consiste de 7 dominios de 3 letras tales como: edu, com y gov. Comúnmente Internet divide al grupo de dominios organizacionales dentro de 7 categorías básicas. La siguiente tabla presenta esta clasificación:

Dominio	*Descripción*
com	Organizaciones Comerciales (negocios)
edu	Organizaciones Educativas (universidades, secundarias, etc.)
gov	Organizaciones Gubernamentales de Estados Unidos
int	Organizaciones Internacionales
mil	Organizaciones Militares de Estados Unidos
net	Una red que no entra dentro de las otras categorías de los dominios (recursos de red)
org	Otro tipo de Organizaciones

Tabla 3.3 Los dominios genéricos u organizacionales básicos

Solamente los dominios gov y mil representan estrictamente organizaciones gubernamentales y militares de Estados Unidos.

3.- El grupo **geográfico** o de país consiste de etiquetas de dominio que representan códigos de países por medio de 2 letras que la Organización Internacional de Estándares (ISO) ha definido en su documento ISO 3166.

En la forma en que Internet debe proporcionar a cada país la responsabilidad de sus propios nombres, para estos días existe un conjunto de códigos de 2 letras que corresponden a los dominios de nivel superior de los países. Por ejemplo: el código *be* es el nombre que identifica a Bélgica, una computadora que se encuentre en Bélgica puede ser nombrada como : *uia.ua.ac.be*. El nombre *servidor.dgsca.unam.mx* representa a una máquina cuyo país de ubicación es México (*mx*).
La siguiente tabla muestra los códigos empleados para identificar a algunos países:

Código	***País***	***Código***	***País***
de	Alemania	la	Laos
ar	Argentina	ls	Lesoto
am	Armenia	lt	Lituania
au	Australia	lu	Luxemburgo
at	Austria	mg	Madagascar
bs	Bahamas	mw	Malawi
bd	Bangladesh	my	Malasia
be	Bélgica	ml	Malí
bz	Belice	mq	Martinica

bo	Bolivia	mx	México
br	Brasil	md	Moldavia
bg	Bulgaria	mc	Mónaco
ca	Canadá	mn	Mongolia
cl	Chile	mz	Mozambique
cn	China	na	Namibia
co	Colombia	np	Nepal
cr	Costa Rica	nl	Holanda
cu	Cuba	nz	Nueva Zelanda
cs	Checoslovaquia	ni	Nicaragua
do	Rep. Dominicana	ng	Nigeria
ec	Ecuador	pk	Pakistán
eg	Egipto	pa	Panamá
sv	El Salvador	py	Paraguay
et	Etiopía	pe	Perú
fi	Finlandia	ph	Filipinas
fr	Francia	pt	Portugal
gr	Grecia	ro	Rumania
gt	Guatemala	ru	Rusia
gn	Guinea	rw	Ruanda
gy	Guayana	sa	Arabia Saudita
ht	Haití	sn	Senegal
hn	Honduras	sg	Singapur
hk	Hong Kong	so	Somalia
hu	Hungría	za	Sudáfrica
in	India	su	Unión Soviética
id	Indonesia	es	España
ir	Irán	sd	Sudán
iq	Irak	ch	Suiza
il	Israel	sy	Siria
it	Italia	tz	Tanzania
jm	Jamaica	uk	Reino Unido
jp	Japón	us	Estados Unidos
jo	Jordania	uy	Uruguay
kr	Corea	ve	Venezuela
kw	Kuwait	vn	Vietnam
lr	Liberia	yu	Yugoslavia
ly	Libia	zr	Zaire

Tabla 3.4 Código de Países en Internet

Responsabilidades

El problema de asignar los nombres es un asunto que se ha delegado a diferentes niveles de la jerarquía del Sistema de Nombres. El Centro de Información de la Red de Internet (InterNIC) maneja el nivel superior de los nombres de dominio y delega la responsabilidad de asignar los nombres subsecuentes a diferentes organizaciones. Cada organización es responsable de una porción específica de la estructura del DNS. Los profesionales de Internet se refieren a estas áreas de responsabilidad como zonas. La organización responsable de una zona específica puede, a su vez, subdividir esa zona y delegar responsabilidades para la asignación de nombres. Esta subdivisión puede continuar hasta que sea una sola entidad la que se encargue de la asignación de nombres dentro de una zona bien definida. A esta entidad individual se le conoce como administrador del DNS y es quien se encarga de dar de alta servidores de nombres para su zona.

El administrador de un DNS no solo se involucra en mantener en ejecución los programas apropiados en los hosts servidores y en los hosts clientes, sino también determinar nombres de dominio, atender quejas, llenar formas de registro y otras formas que lo unan a una red pública. Las responsabilidades de un administrador dependen de la posición de su dominio dentro de la jerarquía del sistema de Internet. Por ejemplo, manejar un grupo de servidores de nombres de un dominio administrativo pequeño encierra menor responsabilidad que el manejar el conjunto con mayor autoridad de una zona más amplia. Las responsabilidades dependen en si el administrador es una autoridad superior para un dominio o zona, o es un administrador que se reporta con una autoridad superior.

El InterNIC divide a los administradores de Internet en *administradores de dominio*, que tienen responsabilidades primordiales para un dominio, y en *contactos técnicos* que trabajan con los administradores de un dominio y mantienen una zona.

Administrador de Dominio: Es un coordinador, administrador, y técnico para un dominio de segundo o bajo nivel. Sus responsabilidades incluyen:

- ☆ Registrar el dominio. El dominio debe registrarse para estar en una red pública, y debe tener un nombre único dentro del nivel de la jerarquía del sistema a la que pertenece . Para registrarse debe de obtener la forma de registro de dominio del InterNIC.
- ☆ Nombrar hosts y verificar que estos nombres sean únicos dentro de dicho dominio.
- ☆ Entender el funcionamiento de los DNS y asegurar que los datos contenidos en ellos estén actualizados

Contacto Técnico: La responsabilidad primordial de un contacto técnico es el de mantener los programas del DNS y los archivos de la zona, así como mantener funcionando los servidores de nombres de la zona . Los contactos técnicos interactuan con los administradores de dominio para resolver problemas en la red.

3.3.2.- SERVIDORES DE NOMBRES

Aunque los usuarios prefieran los nombres de dominio en lugar de las direcciones IP, los programas de aplicación necesitan imprescindiblemente de las direcciones de Internet. Hemos mencionado que un Servidor de Nombres es un programa que traduce de forma fiable y con rapidez los nombres de dominio a sus direcciones IP. Dentro del sistema de Internet, miles de computadoras contienen el software especial para servidores de nombres. Cuando los programas de aplicación necesitan conectarse a un host particular, generalmente primero contactan a un programa servidor de nombres y solicitan el servicio de búsqueda del DNS. Este DNS consulta la información en una estructura de datos que contiene el nombre del host, un alias (si lo hay), la dirección IP en notación decimal con puntos, y la dirección binaria de 32 bits. Un alias puede ser asignado a un nombre para poder referirse a él en una forma más corta, por ejemplo: el servidor *indy.aragon.unam.mx* tiene como alias "*indy*", de manera que cuando un usuario desea hacer referencia a este host, no requerirá escribir el nombre completo, basta con referirse a el como *indy*.

En muchas ocasiones, una organización puede dedicar una computadora en particular para ejecutar el programa servidor de nombres. En tales casos se puede decir que dicha computadora es un servidor de nombres. La tarea de los servicios de búsqueda del sistema DNS es una operación decisiva, ya que al menos que un programa de Internet traduzca nombres de dominio a sus respectivas direcciones IP binarias, no se efectuará comunicación alguna entre redes.

El Servicio de Nombres de Dominio es un protocolo de la capa de aplicación que es parte de la serie de los protocolos TCP/IP. Específicamente, DNS es un servicio de "nombramiento"; este obtiene y proporciona información sobre computadoras hosts de una red.

El Servicio de Nombres de Dominio desarrolla este nombramiento entre computadoras hosts dentro del dominio administrativo al que pertenece una red local y también con aquellas que están al otro lado de las fronteras de dicho dominio. Este servicio es distribuido entre un grupo de servidores, comúnmente conocidos como Servidores de Nombres, cada uno de los cuales implementa el servicio DNS por medio de la ejecución del programa servidor (demonio-daemon) llamado *in.named*.

El demonio *in.named* también es reconocido como el servicio *BIND* (Berkeley Internet Name Domain), que fue desarrollado en la Universidad de California en Berkeley.

Del lado del cliente, DNS es implementado a través del resolvedor. Este resolvedor no es un demonio y tampoco un programa en particular; en cambio, es una librería compilada dentro de aplicaciones que necesitan reconocer nombres de máquinas. La función del resolvedor es el "resolver" peticiones de los usuarios, y para hacer eso, éste consulta a un servidor de nombres, el cual regresa la información solicitada o la referencia a otro servidor de nombres.

Un servidor de nombres corriendo el demonio *in.named* también puede correr el software del *resolvedor*; de ahí que puede haber 2 tipos de clientes:

- solamente hosts clientes: Los host que solamente son clientes no ejecutan el demonio *in.named*; en lugar de eso, estos consultan al resolvedor que proporciona un lista de posibles máquinas servidores de nombres a las que se pueden direccionar las consultas.

- clientes/servidores: Un host cliente/servidor es una máquina que utiliza el servicio de nombres de dominio provisto por el demonio *in.named* para resolver las solicitudes de los usuarios. Sin embargo, si el demonio deja de funcionar, el cliente/servidor tiene la capacidad de resolver peticiones por medio de su resolvedor.

DNS depende de la estructura jerárquica de Internet y agrega a esa estructura el concepto de "zonas de dominio". El sistema de Internet requiere que cada zona cuente con un servidor de nombres primario y uno o más servidores de nombres secundarios (de respaldo). Un servidor de nombres primario almacena toda la información DNS local. Los servidores de nombres secundarios recolectan información del servidor primario de la zona y esto se conoce como transferencia de zona. Con frecuencia, un servidor secundario consultara al servidor primario cada tiempo determinado (algunas horas).

Servidores Primarios Maestros: Estos servidores contienen todos los datos de la zona correspondiente y son la autoridad para esa zona. Estos servidores son conocidos como servidores de nombres superiores o autoritarios. En el servidor primario es donde se hacen los cambios de la zona. Este carga la copia maestra de sus datos del disco cuando inicia el demonio *in.named* y puede delegar autoridad a otros servidores dentro de la zona.

Servidores Secundarios Maestros: Son servidores que mantienen una copia de los datos de la zona. El servidor primario transmite sus datos y delega su autoridad al servidor secundario. Cuando el servidor secundario inicia *in.named*, este solicita todos los datos al servidor primario de la zona dada, y el servidor secundario checara periódicamente con el servidor primario por si existe necesidad de actualizar los datos.

Los servidores de nombres con los que cuenta RedUNAM se describen en la siguiente tabla:

Nombre del Servidor	*Dirección IP*
ns.dgsca.unam.mx	132.248.10.2
danzon.astroscu.unam.mx	132.248.1.3
ns.ans.net	192.103.63.100
nis.ans.net	147.225.1.2

Tabla 3.5 Servidores de nombres en RedUNAM

Un servidor puede funcionar como maestro para múltiples zonas; esto es, como un servidor primario para algunas zonas, y como servidor secundario para otras.

Un servidor del nivel de root de la red es llamado *servidor de nombres del dominio root*. En Internet, los servidores de nombres del dominio de root son mantenidos por el InterNIC.

Tanto el servidor primario como los secundarios no dependen uno del otro. El propósito de usar múltiples servidores de nombres para la misma zona, es el hecho de incrementar la funcionalidad del DNS. Al hacer uso de más de un servidor de nombres por zona, la avería de un sistema individual no hará que el sistema DNS de todo Internet se paralice.

3.3.3.- TRADUCCIÓN DE NOMBRES A DIRECCIONES IP

Para realizar la resolución de un nombre a su correspondiente dirección IP, un cliente (resolvedor) llama al servidor de nombres de la zona. El servidor examina la petición y determina si este tiene la autoridad para el dominio especificado. En caso afirmativo, el servidor traduce el nombre a su dirección IP haciendo uso de la base de datos del servidor de nombres local y después le envía la respuesta al extremo cliente.

Otro caso es, cuando probablemente el servidor local conozca la dirección porque alguien más ha preguntado por ella recientemente. Siempre que se solicita una dirección, el servidor de nombres local la guarda por un momento, por si alguien más desea la misma dirección más tarde, y esto hace que el sistema sea mucho más eficiente.

Cuando el servidor local contactado no puede resolver dicho nombre, el servidor tendrá que contactar con un servidor de root. Este servidor de root es el que conoce las direcciones de los servidores de nombres de la zona de nivel superior.

Los clientes que solicitan la traducción pueden hacerlo de dos formas: *resolución recursiva* y *resolución iterativa*. La primera es la traducción completa, esto es, que si el servidor no puede resolver este nombre, entonces contactará a otro servidor de nombres de dominio que pueda resolverlo y regresara la respuesta al cliente. Por su parte, la resolución iterativa es una petición donde el servidor en caso de no poder resolver el nombre, le indicará al cliente que servidor deberá contactar a continuación. En este tipo de solicitud el servidor no realizará la resolución completa para el cliente.

Ya se ha aclarado que un servidor de nombres no requiere necesariamente de conocer los nombres o direcciones IP de los demás servidores dentro del Sistema de Nombres de Dominio. En cambio, cada servidor de nombres en el DNS debe saber como contactar a los servidores de nombres del nivel de root. La configuración de los archivos para cada servidor primario contiene la dirección IP de cada servidor de nombres de root y estos a su vez deben conocer el nombre y dirección IP de todos los servidores de nombres del nivel secundario. La siguiente tabla señala los servidores de nombres a nivel root del sistema DNS de Internet:

Nombre del Host	*Dirección IP*	*Programa servidor*
NS.NIC.DDN.MIL	192.112.36.4	bind (unix)
AOS.BRL.MIL	128.63.4.82 26.3.0.29 192.5.25.82	bind (unix)
C.NYSER.NET	192.33.4.12	bind (unix)
TERP.UMD.EDU	128.8.10.90	bind (unix)
NS.NASA.GOV	192.52.195.10 128.102.16.10	bind (unix)
NIC.NORDU.NET	192.36.148.17	bind (unix)
NS1.ISI.EDU	128.9.0.107	bind (unix)
NS.ISC.ORG	192.5.5.241	bind (unix)
NS.INTERNIC.NET	198.41.0.4	bind (unix)

Tabla 3.6 Servidores de Nombres de Dominio a nivel root

3.3.4.- OBSERVACIONES SOBRE EL SISTEMA DNS

- Las partes de un nombre típico de dominio describen quién es responsable del mantenimiento de ese nombre. Pero quizá no diga nada sobre quien mantiene la computadora correspondiente a tal dirección IP, o aún (pese al uso de los códigos de país) en donde se encuentra ubicada dicha máquina. Es perfectamente legal tener un nombre : *ns.unet.umn.edu* (parte de una universidad de E.U. en el espacio de nombre) apuntando a una máquina ubicada físicamente en Australia. Esto no es muy común, pero puede suceder.

- Las partes del nombre de un dominio no describen necesariamente a que red pertenece una computadora. Los nombres de Dominio y las redes generalmente se relacionan, pero no existe una coaexión forzosa entre ellos; 2 computadoras con el mismo dominio pueden no estar en la misma red. Por ejemplo, los sistemas *ns.unet.umn.edu* y *nss.unet.umn.edu* aunque están bajo el mismo dominio, pueden localizarse en diferentes redes. Esto significa que los nombres de dominio indican únicamente quien es el responsable del dominio.

- Una máquina puede tener diversos nombres, sobre todo aquellas que ofrecen ciertos servicios, los cuales pueden ser cambiados a diferentes computadoras en el futuro. Por ejemplo, una máquina nombrada como *usd.unet.unm.edu* puede ofrecer el servicio de ftp público dentro de esa universidad y de esta forma también se pueden referir a ella como: *ftp.unet.unm.edu*. Quizá con el tiempo, este servicio sea cambiado a otra máquina y cuando esto suceda, el nombre *ftp.unet.unm.edu* también se moverá junto con el servicio. La computadora principal seguirá con su nombre original *usd.unet.unm.edu*. La gente que requiera de dicho servicio utilizará el mismo nombre sin importar donde se encuentre la máquina que proporcione el servicio. Los nombres que se refieren simbólicamente a un servicio en particular se conocen como "nombres canónicos" (o cnames).

- Los nombres nos son indispensables para la comunicación, y al menos que se reciba un mensaje de "host desconocido", el nombre si es válido. Un mensaje de error de este tipo significa que el sistema no ha podido traducir el nombre dado a una dirección IP.

- Si una computadora que provee un servicio es reubicada de un edificio a otro, su red y, en consecuencia, su dirección probablemente cambiarán. Realmente el nombre no necesita cambiar, cuando el administrador asigne la nueva dirección, lo único que deberá hacer es actualizar el registro del nombre, de manera que dicho nombre apunte a la nueva dirección. Debido a que el nombre aún funciona, el usuario no tendrá que preocuparse si la computadora o el servicio han cambiado de lugar.

Aunque son miles las computadoras que se encuentran en la red, todas y cada una de ellas corresponden a un nombre. La gran ventaja del Sistema de Nombres de Dominio y los servidores de nombres de Internet es el hecho de subdividir la gran Mega-red en módulos que facilitan su manejo y administración.

Capítulo IV
Vías de Conexión

Hoy en día es muy sencillo conectarse a Internet, la "Super Carretera de Información", sin la necesidad de ser parte de una gran corporación, universidad o de tener demasiado dinero. No importa de que persona se trate, lo único que necesita un usuario individual para conectarse a Internet es adquirir una computadora, un modem[11] (mientras más poderosa sea la máquina y más veloz el modem mejor), obtener los servicios de una compañía que sea proveedora de la conexión a Internet y listo.

Los proveedores de estos servicios son demasiados y día a día están participando en un gran mercado de competencia. Por cada tipo de servicio existen diversos proveedores disponibles y como consecuencia diferentes tarifas de los mismos. Para adquirir los servicios de uno de tantos proveedores, la persona interesada debe evaluar: la calidad del servicio contra el costo, el costo inicial contra el costo mensual, el tipo de conexión, etcétera.

Los proveedores de Internet se catalogan en dos grupos, aquellos a nivel nacional y los que son a nivel regional. Los *proveedores nacionales* proporcionan sus servicios a cualquiera que se encuentre dentro de tal nación. Por su parte, los *proveedores regionales* han establecido un área dentro del país y sólo podrán prestar sus servicios dentro de ese límite. Podría pensarse que los proveedores nacionales son quiénes se encargan de realizar la conexión internacional, esto es cierto; Sin embargo, existen proveedores regionales que también lo llevan a cabo. La elección del proveedor más adecuado depende de las necesidades de cada cliente y sobre todo de la vía de conexión utilizada.

Si el cliente es una persona individual o un negocio pequeño, lo más viables es contactar a los proveedores que proporcionan los servicios de *dial-up (marcación) o los servicios de SLIP/PPP*. Las medianas y grandes empresas deben concentrar su atención en la *conexión mediante SLIP/PPP o los servicios dedicados*.

---

[11] *Modem es un dispositivo que conecta a una computadora a una línea telefónica para la transmisión de información.*

4.1.1.- CONEXIÓN DEDICADA A INTERNET

Esta es una conexión permanente y directa a Internet. Esto permite a grandes corporaciones u organizaciones que tengan un acceso completo a la red. El proveedor de este servicio arrienda una línea telefónica con una velocidad de la elección del cliente (mientras más veloz mayor costo), y coloca una computadora o enrutador especial en la ubicación de tal corporación. Dicho enrutador es responsable de llevar a cabo la comunicación entre el sitio local y el extremo requerido (destino), y viceversa. Este servicio tiene un costo muy alto, y al menos que el cliente sea una organización con recursos y que justifique el uso de la red, este tipo de conexión no será de tanto provecho como su gran precio. No obstante del precio, una vez que se ha realizado la conexión, se puede dar acceso a cuanta computadora lo requiera para ser parte de Internet. Para llevarlo a cabo solo es necesario colocar esas computadoras en una red de área local enlazadas con el enrutador.

El acceso dedicado ofrece la conexión más fiable, donde cada computadora es un miembro completo de Internet, capaz de desarrollar cualquier función de red. Sin embargo, debido a que se trata de un acceso muy costoso, esta conexión es más apropiada para grandes corporaciones o empresas y no muy práctica para usuarios individuales.

El acceso dedicado a Internet generalmente requiere una estructura de soporte para la red local. El proveedor del servicio tendrá la obligación de ayudar al cliente al inicio de la conexión, pero una vez que esta queda en marcha, el proveedor sólo se hará responsable del enrutador y la línea telefónica que esta en arrendamiento. De esta forma delega toda responsabilidad de la red local a la corporación en la que se encuentra.

4.1.2.- CONEXIÓN POR LINEA TELEFÓNICA O SLIP/PPP

Este tipo de servicio está tomando mayor popularidad entre los viajeros (usuarios) de la Super Carretera de Información. Esta conexión se efectúa enlazando al usuario con una compañía que tiene una conexión directa con la red, este proveedor habilita a sus suscriptores a comunicarse con la red y hacer uso de esa conexión.

SLIP y PPP son un software creado para habilitar la comunicación a través de líneas telefónicas. Son técnicas muy parecidas al "acceso dedicado" pero con un costo mucho menor. Este software corre bajo líneas telefónicas normales utilizando modems de alta velocidad. El usuario deberá comprar el software SLIP / PPP y un dispositivo modem, pero no sufrirá los altísimos costos de una conexión dedicada. Las ventajas de este servicio es que no se necesita de una línea telefónica dedicada, el usuario utiliza el software y un número telefónico para conectarse a una red cuando requiere accesar a Internet y una vez que termina libera la línea telefónica para su uso normal (llamadas telefónicas, fax, etc.). Además la persona adquiere su propio nombre (hostname) dentro del sistema.

Este tipo de conexión es muy apropiada para que aquellos usuarios que tienen una computadora en casa puedan enlazarse a una red local más grande, que a su vez está conectada a Internet. Esto significa que una persona puede usar este servicio para conectar su computadora personal a la red de su compañía o su universidad (donde ha adquirido una cuenta), y de esta manera la computadora personal tendrá un acceso completo a Internet, podrá hacer transferencia de archivos y navegar por todo el sistema.

Los programas tales como: telnet, ftp, aplicaciones de e-mail, netscape, etc., deben estar instalados en la computadora personal desde donde se hace el enlace.

Como hemos apuntado, los servicios de SLIP / PPP son apropiados para conectar computadoras personales (de casa), y tal vez alguna red local muy pequeña, a un proveedor que pueda proporcionar un enlace completo a Internet. No es muy recomendable para conectar medianas o grandes redes al sistema de Internet, pues no pueden comunicarse lo suficientemente rápido para atender a mucha gente a la vez.

4.1.3.- CONEXIÓN TERMINAL O DIAL-UP

Este tipo de conexión se refiere a que el usuario consiga una cuenta en alguna computadora (servidor) que tiene un acceso dedicado con Internet. Después podrá usar su computadora personal desde su casa para conectarse a ese sistema remoto y realizar su trabajo de red . Este servicio es comúnmente ofrecido por grandes compañías de servicios en línea como Compulink, Delphi o CompuServe. El interesado se suscribe a su sistema que normalmente provee un rango agregado de servicios (tales como conferencias, bases de datos, etc.) y así obtiene acceso a su gateway de Internet. Esto no significa que se tenga una conexión directa, el usuario se conecta a la compañía que si está enlazada directamente a la red. De esta forma, ya no es visto como un "hostname", ahora se visualiza como "usuario@sistema.com.mx" (por ejemplo). Para transferir archivos, primero habrá que bajarlos al sistema en línea (servidor), y después a la computadora personal del usuario. La desventaja de esto es que no se podrá hacer uso de un ambiente gráfico para el WEB y sólo se podrá usar un ambiente de terminal.

En este tipo de conexión al sistema, la computadora del usuario funciona como una terminal de Internet, mientras que la computadora del proveedor de servicios (servidor) realiza la función de un host del sistema Internet. En esta conexión el usuario generalmente accesa a Internet a través de servicios desde un shell de UNIX o un programa que el proveedor ha puesto a su disposición. De manera que cuando el usuario desea ejecutar un programa, por ejemplo un telnet, el programa ejecutable para ese comando reside en la computadora del proveedor del servicio (servidor). En otras palabras, la computadora del proveedor de servicios es la que debe contener todos los programas ejecutables de la serie TCP/IP que el cliente desee ejecutar.

El hecho de que este tipo de conexión es compartido entre otros usuarios, hace que el costo de sus servicios sea muy reducido. El costo de este servicio entre proveedores varía mucho, pero la mayoría están basados en precios por hora. Si se usa mucho Internet, el costo se incrementa, pero es bueno para los que no son usuarios masivos del sistema. Por otro lado, el usuario tiene la ventaja de que posiblemente ya cuente con todo el hardware y software de su computadora requerido (modem, emulador de terminal, etc.), y aunque tuviera que comprarlos, esto no sería de un costo exagerado. La desventaja es que el usuario solo puede realizar las tareas que el proveedor del servicio permite, tal vez no pueda hacer uso de todos los servicios de Internet y en ocasiones los proveedores limitan el espacio de disco que se puede ocupar.

4.1.4.- ACCESO UUCP (Conexión por correo)

Todos los sistemas Unix dan soporte a un conjunto de servicios llamados UUCP (Unix to Unix Copy), los cuales transfieren información sobre líneas telefónicas comunes. Si el usuario encuentra un proveedor de servicio cooperativo (como UUNET), puede enlazarse para hacer uso de los servicios UUCP y tener acceso al correo electrónico de Internet y noticias de USENET (noticias para usuarios de la red). El sistema del cliente utiliza UUCP para marcar a un sistema remoto, y transferir noticias y correo a determinados intervalos de tiempo. De ahí que el usuario pueda leer su correo en su propio sistema y no en el de otros. El usuario no está conectado en realidad a Internet y no se puede hacer mucho más que leer correo y noticias. La computadora sólo marca al sistema remoto de Internet en forma periódica para transferir archivos.

Aunque es probable realizar muchas tareas por medio del e-mail (correo electrónico), generalmente es más difícil y no permite el acceso a los aspectos más importantes de la red. Si todo lo que el usuario desea es tener acceso al e-mail en su sistema casero, lo único que debe hacer es contactar a un sistema que trabaje con UNIX, ya que este cuenta con todo el software requerido para dicho servicio.

Este tipo de servicio puede ser muy barato y en ocasiones la conexión puede ser gratuita.

4.1.5.- ACCESO A TRAVÉS DE OTRAS REDES

La mayoría de las redes de servicios, como Bitnet y CompuServe, han instalado gateways que permiten al usuario el intercambio de correo electrónico con sistemas de Internet. Algunos otros han dedicado gateways que permiten la lectura de boletines de Internet (USENET news), y existen otros pocos esparcidos en el sistema que reciben peticiones de archivos a través de mensajes de e-mail, tales servicios consiguen el archivo y lo mandan por correo automáticamente. Esto no es tan bueno como obtener el archivo directamente (mediante ftp), pero funciona.

Definitivamente esto no representa una conexión a Internet, el usuario solo tiene acceso a pocos servicios, y lo que puede realizar es muy limitado en comparación de todos los servicios con mayor relevancia que el usuario puede encontrar.

4.2.- PROTOCOLOS "SLIP/PPP"

El grupo de usuarios de Internet que crece de manera más rápida pertenece a aquellos que se conectan a la red mediante un proveedor de servicios de Internet y líneas telefónicas. Estos usuarios, como ya se ha descrito, conectan sus modems a líneas telefónicas comunes y después realizan una comunicación de tipo serial para transmitir y recibir información. Cuando esto sucede se emplea uno de dos protocolos que gobiernan la comunicación serial: SLIP o PPP.

4.2.1.- PROTOCOLO SLIP

SLIP (Protocolo de Internet para Líneas Seriales) es un protocolo de arreglos de paquetes. Este protocolo determina una secuencia de caracteres que estructuran a los paquetes IP en una línea serial. SLIP no se enfoca en el direccionamiento, no identifica el tipo de paquetes, y tampoco provee mecanismos de detección/corrección de errores ni de compresión.

SLIP es generalmente usado en enlaces seriales dedicados y algunas veces para conexiones dial-up. Se usa sobre líneas telefónicas con velocidades entre los 1200 bps y 19.2 kbps. Es útil para que grupos de computadoras hosts y enrutadores puedan comunicarse con otras máquinas (host-host, host-enrutador, y enrutador-enrutador).

SLIP está disponible para la mayoría de los sistemas basados en Unix, algunos concentradores de terminales y también para PC's compatibles con IBM.

Para crear una conexión SLIP, un par de computadoras que usan modems y líneas telefónicas establecerán un enlace serial asíncrono. Las dos computadoras transmitirán información mediante el enlace asíncrono en intervalos de tiempo arbitrarios. Desafortunadamente, entre esos intervalos frecuentemente el cable de transmisión adquiere algún ruido electrónico. Como consecuencia, los dispositivos que transmiten mediante enlaces seriales utilizan parámetros en la comunicación para poder distinguir entre los bits de información y los bits de intervalo (o ruidos de línea).

Parámetros

Cuando se emplea un modem y un software de comunicación para intercambiar información con otra computadora, se deben especificar ciertos parámetros como la velocidad en baudios, el tamaño de los datos, la paridad, etc., y para establecer comunicación por medio de una conexión serial, los dos sistemas deben usar la misma configuración. La ***paridad*** es un proceso que las computadoras, modems, y otros dispositivos utilizan para detectar la corrupción de la información. En una conexión serial los datos se transmiten en un flujo de 1's y 0's (bits), donde generalmente un número de 8 bits representan una unidad de datos (byte). Algunos protocolos seriales requieren que el dispositivo de transmisión agregue un bit llamado "bit de paridad" y este protocolo puede ser de

paridad impar o paridad par. Un protocolo de paridad impar establece el valor del bit de paridad de forma que el número de 1's en un paquete siempre sea impar. De la misma forma, un protocolo de paridad par coloca el valor del bit de paridad de manera que el número de 1's del paquete siempre sea par.

Por ejemplo, un modem basado en paridad impar que transmite el byte 10001100, tendrá que colocar un 0 como bit de paridad debido a que este paquete de bits ya contiene un número impar (3) de 1's . De otra forma, si el byte fuera 10001101 (número par de 1's), en este caso el modem tendría que colocar un 1 como bit de paridad para hacer que el total de 1's fuera impar (5).

Para que un modem pueda detectar que ha ocurrido la corrupción de los datos durante la transmisión, se realiza el conteo del número de 1's en el paquete. Si el modem transmisor maneja una paridad impar, entonces el modem que recibe debe contar un número impar de 1's en el paquete recibido (incluyendo el bit de paridad). En caso de que el modem receptor cuente un número par de 1's, entonces sabrá que los datos han cambiado y no son correctos. Cuando los datos recibidos han sido violados, el modem receptor los desecha y solicita una retransmisión.

Dentro de la transmisión asíncrona de los datos, el modem debe saber donde comienzan y donde terminan los paquetes de la información. Diversos protocolos de comunicación emplean un ***bit de inicio*** y un ***bit de fin*** para que el modem pueda interpretar correctamente los paquetes que son recibidos. Un bit de inicio siempre es un 1 e indica a la computadora receptora que se realizará una transmisión en la línea y el siguiente bit pertenecerá a los datos. Por su parte el bit de fin es un 0 e informa el termino de los datos que dejará libre la línea de transmisión.

Los modems trabajan a ciertas ***velocidades en baudios***. Frecuentemente se cree que un baudio representa bits de datos por segundo, es decir, se tiene la idea de que 1200 baudios significan 1200 bits (150 bytes) de datos por segundo (bps). Sin embargo, los modems no solo transmiten paquetes de 8 bits de datos, también se incluyen un bit de inicio, un bit de termino y en ocasiones un bit de paridad. Así, cada paquete de datos contiene de 10 a 11 bits y esto implica que una transmisión de 1200 baudios puede transmitir entre 110 y 120 bytes por segundo. De la misma forma, un modem de 9600 baudios puede transferir entre 872 y 960 bytes por segundo. En el supuesto caso de que un modem trabaje con las nuevas tecnologías de compresión de los datos, el número de bits transferidos puede exceder lo definido hasta en un 200 %.

Establecimiento de Conexión SLIP

Una conexión SLIP es una de las maneras más fáciles y económicas para enlazar una computadora personal PC con Internet. Para establecer una conexión SLIP a Internet es necesario contar con los servicios de un proveedor que pueda conceder cuentas SLIP. El proveedor del servicio se encarga de asignar cuentas y tener uno ó más números telefónicos donde el usuario marque para poder accesar a esas cuentas. Los números telefónicos dependen de la ubicación del proveedor del

servicio, de manera que cuando se haga la elección de un proveedor hay que tomar en cuenta no solo el costo del servicio sino también el costo de las llamadas telefónicas, ya que muchas veces las llamadas de larga distancia suelen ser más costosas que el mismo servicio. Es recomendable comparar los costos entre los proveedores de servicios de Internet considerando las diferencias entre tarifas mensuales y las tarifas por hora en uso de la línea.

Para hacer uso de SLIP se necesita de un software que maneje una conexión SLIP entre la computadora personal y el sistema de Internet. Este software, comúnmente conocido como manejador TCP, es como un manejador de dispositivo de red. Muchos manejadores de SLIP incluyen un programa de marcación (dialer) que se utiliza para llamar al proveedor de Internet.

Normalmente el usuario comienza por cargar el manejador de SLIP en forma interactiva como el lo requiera. Después de cargar el software de SLIP, se pueden correr los programas basados en la serie TCP/IP, tales como Ftp para Windows, Telnet, etc. Para ejecutar esos programas se requiere tener instalada en la PC una copia del programa ejecutable. Es decir, suponiendo que se hará uso del programa Ftp para poder transferir archivos mediante el Protocolo de Transferencia de Archivos, entonces se debe disponer del programa ejecutable para Ftp. De la misma forma, si lo que el usuario desea es leer noticias de la red o usar el correo electrónico, debe contar con el software correspondiente almacenado en su computadora local.

Cuando un usuario establece una conexión SLIP con Internet, su computadora se convierte en un nodo del sistema de Internet. Esto significa que su computadora se vuelve un host con su propia dirección IP. De tal forma, puede cargar y ejecutar sus programas desde su propia PC y esa es la razón de tener almacenados los programas ejecutables en su propia computadora.

Con una conexión SLIP, la computadora no solo se convierte en un host del sistema, sino que ahora también puede ofrecer servicios a otros usuarios de Internet. Por ejemplo, existe el caso de que un usuario con conexión SLIP pueda correr un programa servidor de Ftp en su propia PC y así habilitar que otros usuarios de Internet puedan transferir archivos desde esa PC a las máquinas de ellos. En efecto, un usuario de Australia puede transferir archivos desde una PC en los Estados Unidos a su propia máquina en Australia y todo por el costo de una llamada telefónica a su proveedor local de servicios de Internet.

Estructura de los datos SLIP

En el capítulo 2 se mencionó como las diferentes capas de la pila de protocolos TCP/IP encapsulan los datos en los formatos requeridos por las siguientes capas. Conforme los datos viajan a través de la pila de protocolos, cada capa construye sobre la encapsulación de la capa previa. Para enviar "datagramas IP" a la capa de red, la capa de enlace los traduce en arreglos de datos que sean aceptables para la tecnología de red utilizada. Por ejemplo, la capa de enlace (IP) encapsula los datagramas IP dentro de arreglos de Ethernet para una red Ethernet. Similarmente, la capa de enlace

encapsula los datagramas IP dentro arreglos Token Ring para las redes con tecnología Token Ring.

Cuando en lugar de una tarjeta interfaz de red (Ethernet, Token Ring, etc.) utilizamos un modem para conectarnos a Internet, la capa de enlace es manejada por los protocolos de líneas seriales. El Protocolo de Internet para Líneas Seriales (SLIP) y el Protocolo de Internet para Líneas Seriales Comprimidas (CSLIP) simplemente definen otra capa de encapsulamiento. SLIP y CSLIP preparan los datos para ser transmitidos a través de una interfaz serial (comúnmente RS-232) para enlazarse a Internet. Este software de protocolos se encuentra entre el puerto serial de la computadora y la pila de protocolos de la misma.

El protocolo SLIP es un protocolo que maneja arreglos de paquetes y define la manera como una computadora encapsula los datagramas IP antes de transmitirlos a través de una línea de datos serial. El protocolo es muy simple ya que no tiene la capacidad de direccionamiento, identificación del tipo de paquete, detección o corrección de error, o la compresión de paquetes. Por estas razones, SLIP es muy fácil de implementar y tremendamente popular. Sin embargo, aún no es un estándar oficial de Internet, y por ello, en lugar de SLIP, la mayoría de las compañías utilizan el protocolo PPP (Point-to-Point Protocol) como estándar para la comunicación serial TCP/IP de datos, ya que este sí es un estándar oficial de Internet que especifica un protocolo similar a SLIP para la comunicación serial y de punto-a-punto de los datos.

Arreglos SLIP

Sabemos que cada protocolo encapsula o arregla los datos que recibe, y SLIP no es la excepción. SLIP utiliza una secuencia de caracteres que encapsulan o estructuran cada paquete IP en una línea serial. Este protocolo define 2 caracteres para propósitos de estructuración: *End* (ASCII 192 - 0xC0) y *Esc* (ASCII 219 - 0xDB), un host de SLIP transmite un carácter End al final de cada paquete y utiliza el carácter Esc para marcar los bytes de datos dentro del paquete que tiene el mismo valor como End y Esc. Cuando se encuentra un carácter End, SLIP usa el carácter Esc para avisar al receptor que el valor de End que sigue no es el final de un arreglo. Por ejemplo, si el paquete de los datos incluye un byte con el valor 0xC0 (igual al carácter End), SLIP lo sustituye por la secuencia de escape de 2 bytes *Esc 0xDC* (DB,DC). Si un byte dentro del paquete tiene el mismo valor de Esc, SLIP sustituye la secuencia de escape de 2 bytes *Esc 0xDD* (DB,DD).

La implementación de SLIP en el extremo receptor de una conexión, interpreta los datos en forma opuesta. Esto es, si un host SLIP encuentra el carácter ESC en una corriente de datos serial, SLIP examina inmediatamente el próximo carácter para efectuar los posibles cambios como sigue: cuando SLIP encuentra la secuencia de 2 bytes de escape *Esc 0xDC*, entonces lo sustituye por un byte con el valor de 0xC0 en lugar de los 2 bytes; si SLIP encuentra la secuencia de dos bytes *Esc 0xDD*, los reemplazará por el byte 0xDB. Cuando SLIP observe en una corriente de datos el carácter End sin estar precedido por un Esc, entonces sabrá que ha llegado al final del arreglo. Así, SLIP

pasa todos los datos precedentes a la capa de red como un paquete IP.

En la figura 4.1 se muestra un paquete IP que incluye 2 bytes: uno con valor de Esc y otro con End, y ejemplifica la forma en que SLIP encapsularía dicho paquete.

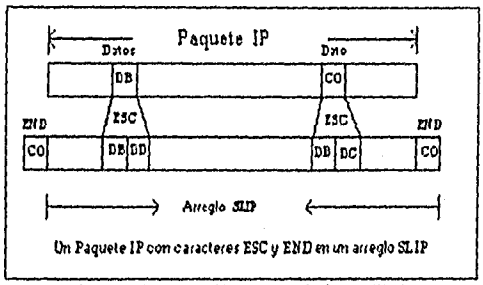


fig. 4.1 Paquete IP encapsulado en SLIP

Phil Karn en el documento RFC1055 sugiere que igual que al terminar un paquete, estos inicien con un carácter END, y de esta forma SLIP elimina cualquier byte erróneo causado por algún ruido telefónico o de las líneas seriales. Las implementaciones de SLIP que emplean estas técnicas pueden desechar un arreglo de longitud cero cuando se detecta dos caracteres End encontrados uno tras otro. La figura 4.2 ejemplifica 2 paquetes IP encapsulados en arreglos SLIP que utilizan la técnica de un carácter End antepuesto al inicio de un paquete IP.

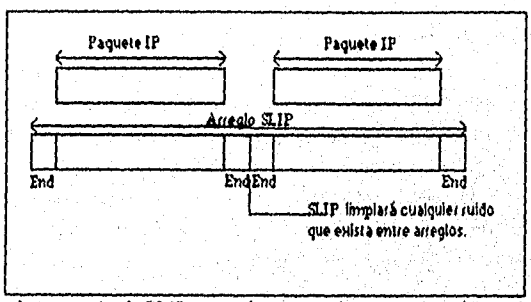


fig. 4.2 Implementación de SLIP que utiliza un carácter interlineal para los arreglos

Como se mencionó anteriormente, al colocar un carácter End al inicio de un paquete IP, SLIP podrá eliminar cualquier byte de erróneo causados por ruido en las líneas de transmisión. No obstante, SLIP no brinda una verdadera detección o corrección de error. En vez de esto, SLIP depende de las capas de TCP/IP de red y de transporte para detectar y descartar paquetes y mensaje inválidos. Por ejemplo, sabemos que IP requiere de un checksum que cubra el encabezado IP. Así, en una conexión SLIP, la capa IP se encargará de detectar el error y desechar el paquete. De forma similar, TCP utiliza un checksum para detectar errores en los encabezados y segmentos TCP, y de esta manera TCP encontrará cualquier corrupción de los datos ocurrida en una conexión SLIP.

Debido a que son los protocolos TCP/IP los que se encargan de detectar y manejar los errores ocurridos, SLIP no se preocupa propiamente de la detección de errores.

Debido a que SLIP no es una especificación estándar, no existe un verdadero límite en el máximo tamaño de un paquete SLIP. Se considera una buena opción el aceptar como máximo tamaño el usado por los controladores de SLIP para Unix con 1006 bytes incluyendo los encabezados IP y del protocolo de transporte empleado (sin incluir los caracteres del arreglo). De ahí que no deberían mandarse datagramas con más de 1006 bytes.

Deficiencias de SLIP

SLIP es un protocolo muy simple diseñado hace mucho tiempo cuando todavía la red no era tan extensa y algunas situaciones no tenían gran importancia.

De lo visto en el capítulo 2, recordemos que ambos protocolos IP y TCP efectúan checksums y se puede confiar en ellos para realizar una conexión SLIP donde se brinde una detección de error en los mensajes enviados por este enlace. En contraste, debido a que UDP no requiere de un checksum, no se tiene una garantía fiable de que los datagramas UDP enviados en una conexión SLIP llegaran correctamente a su destino. Por ejemplo, en el caso de transmitir un datágrama UDP a través de una conexión SLIP y ocurriera el hecho de que ruido en las líneas telefónicas dañara el área de datos del datagrama. Si el datagrama no incluye un checksum UDP, entonces el host receptor no tendrá conocimiento de dicha corrupción en los datos recibidos.

Deficiencias de detección y corrección de error

El ruido de las líneas telefónicas pueden violar los paquetes transmitidos. Debido a que la velocidad de transmisión de la línea no es muy rápida, retransmitir un paquete significaría más gasto. La detección de error no es necesaria en el nivel de SLIP porque cualquier aplicación de IP puede detectar paquetes dañados (encabezados de IP y UDP, y los checksums de TCP deben bastar). Ya que toma tiempo volver a transmitir un paquete que ha sido dañado por el ruido de la línea, sería eficiente que SLIP habilitara por su cuenta algún mecanismo simple de corrección.

Además de la deficiencia en la detección de errores, SLIP es incapaz de direccionar paquetes, identificar diferentes tipos de paquetes, o comprimir la información de los paquetes.

Deficiencias de Direccionamiento.-

Ambas computadoras en un enlace SLIP necesitan saber la dirección IP de la otra para realizar el enrutamiento de los paquetes. Además cuando se establece un enlace SLIP con host que marcan a un enrutador, el esquema de direccionamiento es muy poco dinámico y el enrutador debe informar al host sobre la dirección IP establecida para el mismo.

Este protocolo no proporciona un mecanismo para que los hosts comuniquen información de direcciones por medio de una conexión SLIP.

Debido a que cada proveedor normalmente asigna las direcciones IP de su banco de direcciones disponibles, el usuario puede obtener diferentes direcciones IP con cada conexión SLIP que realice. De esta forma, otros hosts de Internet que quieran accesar los programas servidores de una PC pueden tener dificultad para localizarla.

Además de esto, SLIP no provee método alguno para que el proveedor del servicio indique al software SLIP del usuario cual es su dirección IP. Por ello, cada vez que se establezca una conexión SLIP, el usuario debe indicar manualmente al software SLIP que dirección IP usar. SLIP tampoco ofrece la forma de que un host informe a otro host de su dirección IP. Entonces cada ocasión que se establece una conexión SLIP, el usuario transmitirá manualmente su dirección IP al host de destino. Para poder corregir algunas de estas deficiencias de direccionamiento, se puede establecer una dirección IP dedicada (una dirección que no cambie y pertenezca a un solo usuario) con un proveedor del servicio de Internet. De cualquier forma, los servicios de direcciones IP dedicadas son más costosos que hacer uso de una dirección IP de un banco de direcciones disponibles.

Deficiencias en la identificación del tipo de paquetes.-

La mayoría de las computadoras pueden correr más de una familia de protocolos al mismo tiempo. Por ejemplo, una computadora de la Corporación de Equipos Digitales (DEC) puede correr la serie TCP/IP y los protocolos DECnet. La forma ideal para evitar el incremento de cableado y reducir los costos de hardware, es lograr que múltiples protocolos puedan compartir las mismas líneas de transmisión. Con la tecnología Ethernet, tal compartimiento de cables es posible. Debido a que los arreglos Ethernet contienen un campo "tipo" que define el protocolo de destino del paquete, los paquetes de Ethernet pueden compartir las líneas de transmisión con otros paquetes que tengan campos similares para la identificación de protocolos . Desafortunadamente, un arreglo de SLIP no incluye campo alguno que identifique el protocolo destinatario de dicho paquete. Y esto da como resultado que si dos computadoras DEC corren los protocolos TCP/IP y DECnet paralelamente, no existe la posibilidad de tener a ambos protocolos compartiendo una misma línea mientras se use SLIP.

Deficiencias de Compresión.-

Ya que las líneas seriales suelen ser muy lentas, la compresión de los datos de los paquetes podría incrementar la velocidad de transmisión de los mismos.

Aunque las redes Ethernet pueden transmitir hasta 10 millones de bits por segundo, una conexión SLIP puede incluir modems de alta velocidad que transmitan datos hasta 19,200 bits por segundo. En otras palabras, Ethernet tiene una velocidad 500 veces más rápida que una conexión

SLIP. Para incrementar relativamente la baja velocidad de transmisión de SLIP, se puede llevar a cabo la compresión de datos, que reduce la cantidad de información que deberá transferir una red. Al comprimir los datos, efectivamente se transmite mayor información en menor tiempo.

Por ejemplo, supongamos que se desea hacer la transferencia de un paquete con 500Kb a través de un modem de 9600 baudios. El modem podrá transmitir 960 bytes por segundo aproximadamente y para transferir los 500Kb se requerirán un poco más de 533 segs. (8.8 minutos):

Entonces hagamos la operación, si el modem transmite 960 bytes por segundo, cuántos segundos requerirá aproximadamente para transmitir 500 kb (500 * 1024 bytes) ?

960 bytes = 1 seg.

512000 bytes = ?

512,000 bytes / 960 bytes por segundo = 533.33 segundos

533 segundos / 60 segundos por minuto = 8.8 minutos

Sin embargo, si se comprimen los datos un 25%, se puede reducir la cantidad de datos desde 500Kb hasta los 125Kb. Como resultado, se reduce el tiempo de transmisión a menos de 3 minutos.

125	** 1024 bytes*	*= 128000 bytes*
128000 bytes	*/ 960 bytes por segundo*	*= 133.33 segundos*
133.33 segs.	*/ 60 segundos por minuto*	*= 2.2 minutos*

Muchos modems proporcionan técnicas adheridas para la compresión . De igual forma, algunos protocolos incluyen software de soporte para la compresión de datos. Frecuentemente los flujos de paquetes en una sola conexión TCP contienen pocos campos que son modificados en los encabezados IP y TCP, así que algoritmos de compresión simple podrían enviar solamente los datos de aquellos campos modificados de los encabezados en lugar de los encabezados completos. El protocolo SLIP, no incluye tipo alguno para la compresión de datos. Sin embargo, SLIP Comprimido (CSLIP) si comprime encabezados de paquetes IP y segmentos TCP para incrementar la transmisión.

4.2.2.- COMPRESSED SLIP (CSLIP)

CSLIP comprime solamente la información de los encabezados de IP y TCP en los segmentos TCP y no los datos en sí. Cabe aclarar que CSLIP no comprime los encabezados UDP ni encabezados IP en los datagramas UDP.

Como cualquier otra conexión de red, una conexión de línea serial como SLIP o CSLIP manejan paquetes de datos que incluyen un encabezado del paquete y la información del usuario. Para incrementar la cantidad de información del usuario que transfiere la conexión, se puede hacer uso de la compresión de datos y reducir el tamaño del encabezado del paquete. Se puede dividir la transferencia de datos de una red en dos categorías básicas: *datos interactivos* y *datos de transferencia en volumen.*

FTP y NNTP (Protocolo de Transferencia de Noticias de la Red) son ejemplos de programas que desarrollan transferencia de datos por volumen. Aunque se puede iniciar ambos procesos de manera manual, la mayoría de la transferencia de datos no ocurre interactivamente. Por ejemplo, cuando se transfiere un archivo desde un host FTP, se especifica el nombre del archivo a transferir y se inicia la transmisión. El volumen de datos transferidos son los bytes de datos que pertenecen al archivo. Similarmente, con un programa para noticias de la red, el usuario selecciona interactivamente el grupo de noticias a transferir. Sin embargo, la transferencia de datos que resulta no requiere mayor acción por parte del usuario- no es interactivo.

En forma opuesta, una aplicación basada en Telnet es un ejemplo de la transferencia de datos interactiva. Comúnmente, cada tecla oprimida en un programa de Telnet produce un paquete de datos. Aunque Telnet provee los medios para enviar líneas enteras al mismo tiempo, la mayoría de las implementaciones todavía mandan un paquete por tecla y el host receptor produce un eco de vuelta sobre cada tecla a través de la red hacia la computadora del usuario. Además sabemos que los programas basados en TCP (el caso de Telnet) requieren mensajes de reconocimiento ACK. Esto implica que, un programa interactivo como Telnet genera una cantidad masiva de números de pequeños paquetes de datos.

En los capítulos anteriores se describió que los encabezados IP y también los de TCP normalmente tienen una longitud de 20 bytes cada uno. De esta forma, un programa basado en Telnet genera largas cantidades de segmentos TCP que contienen 40 bytes de información de encabezado por cada byte de datos transmitidos.

Para aumentar la eficiencia de la línea, simplemente se debe incrementar los datos por paquete o decrementar el tamaño del encabezado del paquete. La metodología del protocolo CSLIP está enfocada a una buena respuesta de interacción desde la inter-red TCP/IP.

Líneas seriales muy lentas también pueden ocasionar que el usuario califique un programa como lento, aún cuando el desarrollo del programa sea verdaderamente rápido. Además de esto, también existen factores de hardware como el ancho de banda que manejan los modems.

Existen límites teóricos en materia de comunicaciones para los anchos de banda que deben emplear los modems sobre líneas telefónicas normales. Un ancho de banda efectivo es aquel que resulta de implementaciones para incrementar la transferencia de datos en forma que parezca como si se excedieran dichos límites. Por ejemplo, cuando se realiza la compresión de la información, se impulsa un ancho de banda efectivo. En otras palabras, la compresión produce mayor transferencia de datos en el mismo período de tiempo. En algunos casos, el ancho de banda efectivo puede exceder los límites teóricos de los canales de comunicación.

Para poder reducir el encabezado TCP/IP de 40 bytes dentro de un paquete hasta un promedio de 3 a 5 bytes, existe un método (RFC1144, *Van Jacobson*) donde la teoría principal es la siguiente:

Jacobson señala que aproximadamente la mitad de la información contenida en el encabezado TCP/IP permanece constante durante una conexión TCP. Después de que los hosts correspondientes establecen una conexión TCP, el protocolo CSLIP requiere que tanto el transmisor como el receptor retengan una copia del último encabezado recibido mediante dicha conexión. CSLIP sustituye un pequeño identificador de conexión que las computadoras utilizan para identificar cada conexión. El protocolo CSLIP transmite cambios que sean requeridos y la computadora host actualiza la información del encabezado que almacena localmente. Mediante cada paquete CSLIP llega, el software de la red examina el identificador de la conexión CSLIP y regraba la información del encabezado previamente recibida. En otras palabras, después de establecer una conexión, CSLIP no transmite información del encabezado cuyo contenido no ha cambiado. Este sencillo paso reduce el tamaño del encabezado TCP/IP a 20 bytes.

CSLIP depende del protocolo de la capa de enlace para indicar al receptor la longitud de un mensaje recibido. Haciendo esto, CSLIP elimina otros dos bytes desde el encabezado TCP/IP (campo de long. total en el encabezado IP). No obstante, después de eliminar el campo de Long. Total, el checksum del encabezado-IP es la única parte esencial que permanece en el mismo. Visiblemente no existe razón alguna para transmitir un checksum de información que no se encuentra dentro del paquete. Además CSLIP necesita que el receptor realice checksums de paquetes sin comprimir, pero para aquellos paquetes comprimidos, CSLIP regenera un checksum local. Esto significa que se eliminan 2 bytes más del encabezado TCP/IP.

De todo lo anterior resulta que queden únicamente 16 bytes de la información del encabezado que pueden cambiar mientras exista una conexión TCP. Es importante aclarar que esos 16 bytes no cambian completamente en la transmisión de cada paquete. Por ejemplo, la transferencia de datos que utiliza el protocolo Ftp solo cambia el ID del paquete, número de secuencia, y el

checksum en la dirección del transmisor-a-receptor. Y el ID del paquete, mensaje de reconocimiento, y checksum, cambian en la dirección de receptor-a-transmisor. El CSLIP transmisor siempre retiene una copia del último paquete enviado y de esta manera, el transmisor conoce cuales campos deben modificarse en el paquete actual. Si el transmisor solo envía los campos que son distintos, el esquema de la compresión reduce el promedio del tamaño del encabezado hasta aproximadamente 10 bytes.

Jacobson apunta que el número ID de un paquete generalmente surge de un contador que se incrementa en uno por cada paquete enviado. Esto significa que la diferencia entre el ID de un paquete y el ID del paquete anterior es un pequeño entero positivo, usualmente menor a 256 (un byte); y frecuentemente igualado a uno. Además, desde el lado transmisor en una transferencia de datos, el número secuencial de un paquete será el número secuencial del paquete anterior más la cantidad de datos en el paquete previo. El tamaño máximo de un paquete IP es de 64,000 bytes. Esto indica que el número secuencial debe ser menor a 2 bytes. En otra palabras, al enviar únicamente las diferencias en los campos modificados en lugar de los mismo campos, CSLIP salva otros 3 o 4 bytes por cada paquete.

Así CSLIP, al transmitir solo los campos modificados de un encabezado, los programas reducen el tráfico en la red y se puede reducir el tamaño del encabezado del paquete para mejorar la transmisión.

4.2.3 .- PROTOCOLO PPP (POINT-TO-POINT PROTOCOL)

El Protocolo de Punto-a-Punto (PPP) resuelve las deficiencias del protocolo SLIP (direccionamiento, identificación de paquetes, y compresión) y además si es un estándar oficial de Internet. Por esta razón, aquellos vendedores y negocios que quieren adoptar un estándar oficial para el enlace serial de datos sobre Internet, se enfocan al uso del protocolo PPP, el cual consiste de tres componentes básicos:

- Un método de encapsulación que permite al software de la red utilizar un simple enlace serial para múltiples protocolos.
- Un Protocolo de Control de Enlace (LCP) que emplea el software de la red para establecer, configurar y probar la conexión de enlace serial. Ambos extremos de la conexión PPP utilizan LCP para negociar las opciones de la conexión.
- Una familia de Protocolos de Control de la Red (NCP's) que permiten a las conexiones PPP usar diferentes protocolos de la capa de red.

Encapsulamiento de PPP

Una de las ventajas de este protocolo es el hecho de ser un método de encapsulamiento que permite al software de las redes utilizar un sólo enlace serial para múltiples protocolos.

Para el desarrollo de este Protocolo PPP, los diseñadores se basaron en una estructura de arreglo internacionalmente aceptado (ISO). El Estándar Internacional 3309 de ISO define un protocolo sencillo para la capa de enlace llamado "Control de Enlace de Datos de Nivel Superior" (HDLC). HDLC usa caracteres especiales como banderas para marcar el inicio y final de los arreglos, parecido como SLIP utiliza el carácter End. Además incluye un campo de Chequeo de Redundancia Cíclico (CRC) para detectar errores en el arreglo. La figura 4.3 que sigue nos muestra el formato de un arreglo PPP.

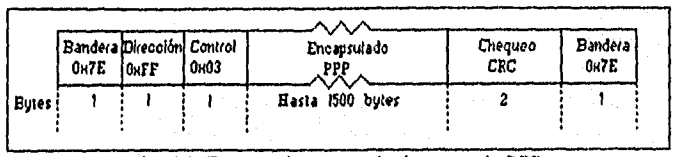


fig. 4.3 Formato de un arreglo de protocolo PPP

Como se puede observar, cada arreglo PPP comienza y termina con un byte de bandera cuyo valor es siempre 0x7E. Los campos de Control y de Dirección utilizan valores compuestos como 0x03 y 0xFF, respectivamente.

Para poder identificar la información dentro de un paquete PPP, existe una estructura básica. La figura 4.4 siguiente muestra la estructura de un paquete dentro de un arreglo PPP.

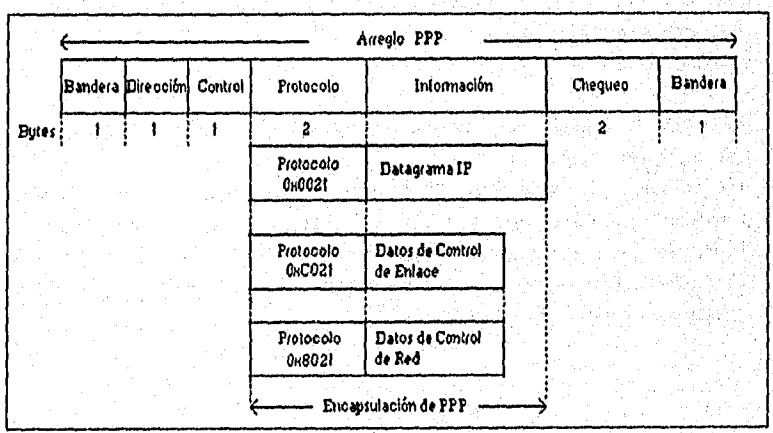


fig. 4.4 La encapsulación en el protocolo PPP

Los dos primeros bytes en la encapsulación PPP (el campo de protocolo) identifican el tipo de datos que contiene el área de información dentro del arreglo PPP . Un campo de Protocolo con valor de 0x0021 especifica al software de la red que los siguientes datos son parte de un datagrama IP. En otras palabras, indica que el paquete PPP contiene el tipo de datos que generalmente fluyen a través de una red TCP/IP.

Similarmente, un campo de Protocolo con valor 0xC021 identifica datos de control de enlace (LCP - Link Control Protocol). Este Protocolo establece, configura, y verifica la conexión de enlace-de-datos. Un campo de Protocolo con un valor de 0x8021 identifica datos de control de red (NCD - Network Control Data). PPP utiliza la información del protocolo NCD para escoger y configurar uno o más protocolos de la capa de red, tal como IP.

El campo de Protocolo de PPP puede ser de 1 a dos bytes de longitud. Para eliminar un byte de cada arreglo PPP (e incrementar la transmisión), las implementaciones PPP generalmente utilizan el Protocolo de Control de Enlace (LCP) para negociar el tamaño del campo de Protocolo a un byte.

Para reducir el tamaño del arreglo por otros 4 bytes, las implementaciones PPP pueden negociar (utilizando LCP) para eliminar los campos de Banderas, Dirección, y Control. Y de esta manera PPP agrega solo tres bytes: uno para el campo de Protocolo y dos bytes para el campo CRC. Además, las implementaciones TCP/IP de PPP pueden emplear el Protocolo de Control de Red (NCP) para negociar el uso de la compresión CSLIP. Básicamente PPP brinda las siguientes ventajas:

- PPP puede soportar múltiples protocolos sobre un simple enlace serial, utilizando el campo Protocolo.
- Empleando el CRC en cada arreglo PPP, el protocolo PPP provee un chequeo de error.
- Por medio del Protocolo de Control de Red, PPP puede negociar la compresión del encabezado TCP/IP.
- Las conexiones PPP pueden negociar nuevas opciones de datos de enlace utilizando el Protocolo de Control de Enlace. Esto permite que los vendedores extiendan PPP sin tener que redefinir el protocolo conforme las tecnologías para enlaces de datos mejoran.

Antes de que las computadoras hosts puedan usar PPP para comunicaciones de red normales, el software de la red debe configurar y probar el enlace de datos. El Software de la red utiliza el Protocolo de Control de Enlace (LCP) para desarrollar las pruebas de enlaces de datos. Después de que los hosts configuran y prueban el enlace de los datos, utilizan el Protocolo de Control de Red (NCP) para elegir y configurar uno o más protocolos de la capa de red, como el caso de IP. Una vez que las negociaciones de LCP y NCP han sido completadas, el enlace de datos queda activo hasta que cualquiera de los protocolos LCP o NCP cierren la conexión. El diagrama 4.5 ilustra el proceso de enlace de PPP.

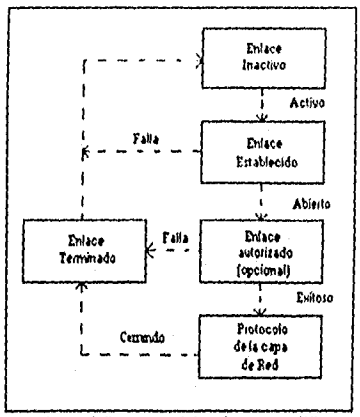


fig. 4.5 Proceso del enlace en PPP

Las fases del proceso de enlace de PPP son :

Enlace inactivo.- Un enlace PPP siempre comienza y termina con la fase de enlace inactivo. En esta fase, la capa física de la red no esta preparada para la comunicación de datos. PPP cambia de la fase de enlace inactivo a la fase de enlace establecido solamente después de recibir alguna indicación de que la capa física de la red esta lista para ser usada. Por ejemplo, en una conexión dial-up PPP interpretaría la señal de detección de acarreo del modem como una indicación de que la capa física esta lista para la transmisión.

Enlace establecido.- Una vez que PPP recibe la indicación de que la capa física de la red esta lista para la comunicación, el protocolo cambia a la fase de enlace establecido. Durante esta fase, el software del módulo PPP en cada host utiliza el Protocolo de Control de Enlace (LCP) para establecer y configurar el enlace de los datos. PPP asume los valores por default para todas las opciones al menos que los hosts determinen alterar una opción durante esta fase. Además PPP negocia opciones independientes para cada red en particular, tales como eliminar los bits de la bandera de no-cambiar en el arreglo PPP. Las implementaciones PPP deben ignorar o descartar aquellos paquetes que no son de LCP (paquetes con un campo de protocolo diferente a 0xC021) recibidos durante esta fase.

Enlace autorizado.- Después de que PPP establece el enlace de datos, este entra a la fase de autorización de enlace. Autorización se refiere al proceso mediante el cual las redes determinan si un host tienen los privilegios necesarios (autorización) para establecer la comunicación con otro host. En otra palabras, una red puede restringir las comunicaciones entre hosts o usuarios de computadoras host. Las redes que transportan información confidencial comúnmente tienen un procedimiento de construcción de autenticidad. Por ejemplo, si una compañía almacena información de contabilidad en una computadora que corre un programa del servidor. Aunque algunos empleados en el departamento de contabilidad de la compañía pueden accesar a esta información, la compañía

no desea que cualquier persona pueda utilizarla. Como resultado, algunas redes incluyen un protocolo de autenticidad (autorización) dentro del servidor de archivos.

Protocolo de capa de red.- En esta fase, el protocolo PPP utiliza los Protocolos de Control de Red (NCP) para configurar uno o más protocolos de la capa de red, tal como IP. Después de que PPP configura los protocolos de la capa de red, la conexión PPP está lista para la comunicación de datos normal y pueden acarrear los paquetes a dichos protocolos. Sabemos que PPP soporta más de un protocolo sobre un simple enlace de datos. Así, mientras la conexión PPP este activa, los Protocolos de Control de Red pueden abrir y cerrar conexiones de protocolos específicos. Por ejemplo, si un módulo de software de la red establece una conexión PPP y configura IP utilizando el Protocolo de Control de Red IP. Más tarde, mientras el enlace de datos IP aún está activo, el módulo del software de la red puede usar la misma conexión PPP para abrir un enlace de datos DECnet. De esta forma, el módulo del software de la red puede terminar el enlace de datos DECnet sin la necesidad de cerrar el enlace de datos IP.

Esto indica que, durante la fase de Protocolo de la Capa de Red, PPP utiliza los Protocolos de Control de Red para abrir, configurar, y cerrar conversaciones de red que utilizan múltiples protocolos de la capa de red. Para configurar cada uno de los protocolos de la capa de red, PPP usa un Protocolo de Control de Red diferente. Por ejemplo, el Protocolo de Control de Red IP es diferente del Protocolo de Control de Red DECnet. Los profesionales en redes diseñan cada Protocolo de Control de Red para manejar los requerimientos peculiares de su correspondiente capa de red.

Enlace terminado.- En esta fase de enlace terminado, PPP cierra la conexión. Si PPP falla en la autenticidad (autorización) de los hosts. PPP entra a la fase de termino de enlace. De igual forma la perdida de la señal de acarreo del modem puede ocasionar que PPP cambie a la fase de terminación. Frecuentemente, el Protocolo de Control de Enlace negocia una terminación de conexión, y es PPP quién debe notificar al protocolo de la capa de red (usando el Protocolo de Control de Red apropiado) sobre el cierre de la conexión PPP. Ya se ha dicho que PPP permite que NCP abra y cierra conexiones de protocolos mientras el enlace de datos este activo. El hecho de cerrar todas las conexiones NCP no es una razón suficiente para cerrar el enlace PPP. Esto significa que, PPP no cierra automáticamente el enlace sólo porque no existan datos fluyendo a través de la conexión. Es el software de la red el que debe terminar específicamente un enlace PPP.

Protocolo de Control de Enlace (LCP)

PPP utiliza el protocolo LCP para negociar y expandir la lista de opciones para configuración de una variedad de ambientes. PPP puede trabajar entre hosts que han sido configurados de forma diferente, y esto debido a que LCP negocia automáticamente las opciones de configuración. Por ejemplo, PPP usa LCP para acordar los formatos de encapsulación, donde puede eliminar campos dentro de los arreglos PPP reduciendo los tamaños de estos arreglos y así poder incrementar la transmisión PPP. Mediante dichas negociaciones, LCP puede hacer dinámicamente cambios en las configuraciones basados en los estados actuales de la red (flujo de transmisión, etc.),

que son transparentes para el usuario. Esto implica que LCP puede establecer configuraciones óptimas.

Existen tres clases de paquetes LCP: de configuración, terminación y mantenimiento. PPP emplea los paquetes de configuración de LCP para establecer y configurar un enlace PPP. Los paquetes de terminación terminan un enlace PPP, y los paquetes de mantenimiento son usados para manejar y depurar un enlace de datos PPP. Cuando el valor en el campo de Protocolo dentro de un arreglo PPP es igual a 0xC021, se identifica que el área de información pertenece a datos del protocolo LCP. La figura siguiente muestra el formato general de un paquete LCP dentro de un arreglo PPP:

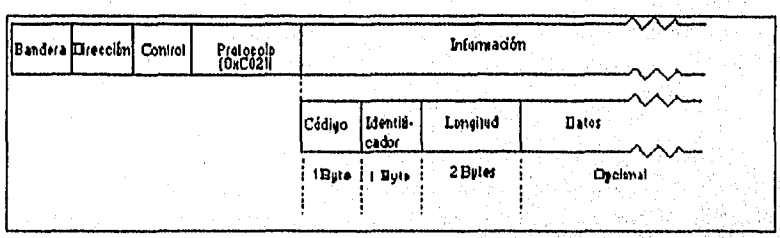


fig. 4.6 Formato del paquete del Protocolo de Control de Enlace (LCP)

El campo de Código identifica el tipo de paquete LCP que se incluye en dicho arreglo. El campo de Identificador enlaza peticiones con respuestas entre las capas de red de una conexión PPP. Este campo de identificación es muy similar al campo de Número de Secuencia que utiliza el protocolo TCP. El campo de Longitud indica la longitud total del paquete LCP, incluyendo los campos de Código, Identificador, Longitud y de los Datos. Por último, el campo de Datos puede ser de 0 bytes (vacío), y es en el campo de Código donde se establece el formato y contenido del campo de Datos. La tabla que sigue señala los códigos establecidos para LCP:

Código	*Nombre del Paquete*	*Clase*
1	Petición de Configuración	Configuración
2	Ack-de-configuración	Configuración
3	Nack-de-configuración	Configuración
4	Rechazo de configuración	Configuración
5	Petición de terminación	Terminación
6	Ack-de-terminación	Terminación
7	Rechazo de código	Mantenimiento
8	Rechazo de protocolo	Mantenimiento
9	Petición de eco	Mantenimiento
10	Respuesta de eco	Mantenimiento
11	Petición descartada	Mantenimiento

Paquetes de Configuración:

Dentro de estos paquetes existen 4 tipos: petición de configuración, Ack-de-configuración, Nak-de-configuración, y rechazo de configuración. El protocolo PPP requiere que siempre se transmita una petición de Configuración para poder abrir una conexión PPP. El campo de datos en un paquete de Petición de Configuración contiene una lista de las opciones deseadas para la configuración.

Cuando un host recibe un paquete con Petición de Configuración, debe de transmitir una respuesta apropiada. Si todas las opciones de la lista en el campo de datos son aceptadas, el host envía un paquete de Ack-de-configuración (reconocimiento). El campo de Datos dentro de un paquete de reconocimiento contiene una copia exacta de todas las opciones de configuración requeridas. En otras palabras, los paquetes de reconocimiento informan que las opciones para configuración han sido aceptadas.

Cuando un host no puede soportar alguna de las opciones requeridas, entonces transmite un paquete de Nak-de-configuración (opciones de configuración no soportadas). El campo de los Datos de este paquete incluyen solamente las opciones de configuración no reconocidas.
PPP seguirá enviando y recibiendo paquetes de Petición y de Configuración No-Reconocidas hasta que ambos extremos de la conexión PPP acuerden las mismas opciones para la configuración. Si los módulos de PPP reciben un paquete con opciones desconocidas, deben transmitir un paquete de Rechazo de Configuración. Este último paquete tendrá en el campo de los Datos solamente las opciones rechazadas. La diferencia entre las opciones no reconocidas en un paquete Nack y las opciones rechazadas en un paquete de rechazo es que las últimas son opciones no negociables.

Paquetes de Terminación:

Son dos tipos de paquetes: los de Petición de Término y los de Ack-de-Término (reconocimiento). Ninguno de los dos utiliza el campo de los Datos del paquete LCP. Cuando un host desea terminar una conexión PPP primero deberá enviar una Petición de Término y continuará transmitiendo estas peticiones hasta que ocurra uno de los tres eventos siguientes:

.- El host reciba un paquete de Ack-de-término (reconocimiento).
.- La capa más baja de la red indique que ya no seguirá disponible para la comunicación
.- Los hosts transmitan los suficientes paquetes de Petición de Término sin respuesta de manera que PPP este seguro de que el otro host del otro extremo está dado de baja.

Paquetes de Mantenimiento.-

Estos paquetes son usados para manejar y depurar los enlaces de datos de PPP. Este Protocolo establece 5 tipos de paquetes de mantenimiento: Rechazo de Código, Rechazo de Protocolo, Petición de Eco, Respuesta de eco, y Petición de Descartar . Cuando un módulo recibe un paquete con un valor irreconocible en el campo de Código, el módulo transmitirá un paquete de Rechazo de Código. De la misma forma, PPP transmite un paquete de Rechazo de Protocolo en respuesta a un valor irreconocible en el campo de Protocolo.

Utilizando los paquetes de Petición de Eco y Respuesta de Eco, PPP puede probar ambas direcciones de un enlace de datos. Cuando un módulo PPP recibe una Petición de Eco, deberá transmitir una Respuesta de Eco. Por su parte, el paquete de Petición de Descartar permite al módulo PPP que verifique el enlace de datos en una sola dirección (desde el host local al host remoto).

Protocolo de Control de Red IP

PPP incluye una serie de Protocolos de Control de Red (NCP) que permiten a las conexiones PPP utilizar diferentes protocolos de la capa de red. En general, el Protocolo de Control de IP (IPCP), habilita y deshabilita los módulos de protocolo IP en ambos extremos o pares de un enlace punto-a-punto.

Con algunas excepciones, el Protocolo de Control IP es muy similar al Protocolo de Control de Enlace (LCP). Sin embargo, PPP encapsula solo un paquete de tipo IPCP en el campo de Información de un arreglo PPP y además el valor del campo de Protocolo para IPCP es 0x8021. IPCP utiliza solamente los primeros siete códigos que define LCP y trata a los códigos restantes como irreconocibles lo cual deriva la transmisión de un paquete de Rechazo de Código.

IPCP define dos tipos de opciones de configuración: Protocolo de Compresión IP, y Dirección IP.

Como ya sabemos, el método CSLIP puede reducir el tamaño de los encabezados TCP/IP. La opción de Configuración de Protocolo de Compresión IP permite a PPP que negocie el uso de un protocolo específico de compresión, como CSLIP. Además esta opción incluye un campo de dos bytes para el Protocolo de Compresión IP y actualmente el único protocolo de compresión identificado por IP es el 0x002D (CSLIP).

Por otro lado, hemos visto que cada vez que establezca una conexión SLIP, el usuario debe de configurar la dirección IP local de su computadora, al menos que su proveedor le asigne una dirección IP permanente. Por el contrario, PPP ofrece a los usuarios de redes TCP/IP la capacidad de negociar direcciones IP. La opción de Configuración de dirección IP para el Protocolo de Control IP habilita a PPP para que solicite una dirección IP específica. Para ello, lo primero que hace es transmitir un paquete de Petición de Configuración con la opción de Configuración de Dirección IP y especificar la dirección IP deseada.

Quizá el hecho de poder configurar la dirección IP sea una de las ventajas más significativas de PPP en comparación al protocolo SLIP.

Capítulo V

Principios Básicos de Telnet, Ftp y E-mail en La Super Carretera de Información

Como lo hemos mencionado desde el inicio de este trabajo, la gran Carretera de Información está disponible a cualquier persona, organización o empresa que se encuentren conectados a la mega-red . La mayoría de estos usuarios recurren a un proveedor que les permita accesar a los servicios de Internet con el software (protocolos) y hardware correspondientes al tipo de conexión. Cada día que pasa, un mayor número de proveedores de hardware, de software, así como de servicios, anuncian nuevos productos enfocados a la explotación de los recursos de información de la red. Sin embargo, *el correo electrónico, las sesiones remotas y la transferencia de archivos* siguen siendo algunas de las herramientas mayormente utilizadas y difundidas en el mundo de Internet.

Toda computadora enlazada al sistema de Internet y debidamente configurada puede tener acceso a una gran diversidad de herramientas y servicios que ofrecen al usuario una amplia variedad de información que va desde disciplinas educativas, científicas, culturales, de investigación, y hasta entretenimiento.

En este capítulo revisaremos los principios básicos de algunas de estas herramientas o servicios con mayor reconocimiento dentro del Sistema de Internet: *Telnet, Ftp y E-mail*.

5.1.- SESIONES REMOTAS < Telnet >

Cuando hablamos de Telnet, estamos hablando de establecer una sesión de trabajo en un host remoto (diferente al local) como si nuestra computadora fuera una terminal enlazada directamente a la red del host remoto. Cuando el usuario se enlaza remotamente, es como si estuviera sentado enfrente de dicha máquina y puede accesar a los servicios que ese host autoriza a sus terminales locales.

Una conexión Telnet es una conexión "TCP" (Transport Control Protocol) utilizada para la transmisión de datos con un control de información bien determinado.

Este protocolo está construido sobre tres ideas principales:

- primero, el concepto de una "terminal virtual de red" (NVT),
- segundo, el principio de negociar opciones y
- tercero, una vista simétrica de procesos y terminales.

NVT (Network Virtual Terminal)

Al iniciar el establecimiento de una conexión de Telnet , cada extremo de dicha conexión esta supuesta a originar y terminar en una Terminal Virtual de Red o NVT. Este término se refiere a un dispositivo imaginario que proporciona un estándar en la representación de una terminal canónica.

No todos los sistemas de computadoras interpretan de la misma forma ciertos grupos de caracteres y otras operaciones de "display" (retorno de carro, cambio de página, tabuladores, backspaces, etc.). En muchas redes, la capa de presentación (modelo ISO/OSI) se encarga de minimizar y resolver la incompatibilidad entre computadoras. Bastantes de estas redes usan protocolos de terminal virtual para desaparecer las diferencias de video e impresión entre sus dispositivos.

La serie de protocolos TCP/IP incluye un dispositivo NVT que funciona como una interfaz estandarizada entre diferentes unidades de display de video o terminales. Este dispositivo funciona como parte esencial del protocolo Telnet y engloba todas las caracterlsticas que normalmente se encuentran en monitores o terminales como VT100. El término NVT define como se lleva a cabo el envío de datos y de comandos, tales como borrar caracteres, crear nuevas líneas, y cambio de carro a través de la red. NVT establece un patrón común para dichas funciones y esconde las diferencias entre las computadoras enlazadas a Internet.

La figura siguiente muestra la función de NVT :

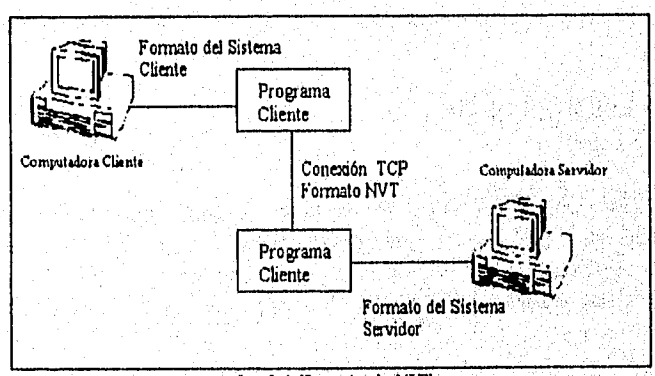


fig. 5.1 Función de NVT

Antes de que ambos extremos transmitan datos o comandos mediante Internet, primero deben traducir esa información a un formato estándar especificado por NVT. Cuando los programas reciben esas transmisiones, ellos utilizan las definiciones de NVT para descifrar y traducir la información recibida al formato utilizado por la computadora local.

El formato del prototipo NVT se basa en el código estándar US_ASCII (de 7 bits) para codificar cartas, dígitos y marcas de puntuación. De los 33 códigos de control que maneja ASCII, NVT solamente reconoce los siguientes ocho:

Código de Control	*Hexadecimal*	*Descripción*
NUL	0x00	Sin operación o efecto
BEL	0x07	Crea una señal audible (bip)
BS	0x08	Mueve a la izq. una posición de carácter (backspace)
HT	0x09	Mueve a la der. al sig. tabulador horizontal
LF	0x0A	Avance de línea
VT	0x0B	Mueve abajo al sig. tabulador vertical (hogar)
FF	0x0C	Mover al inicio de la sig. página (formfeed)
CR	0x0D	Mover al margen izq. de la línea actual (retorno de carro)

Usa de los códigos de control de ASCII en NVT

NVT también define la combinación de CR y LF en el término CRLF para referirse a la operación de la tecla RETURN o ENTER. Generalmente se hace referencia a todos estos términos como NVT ASCII.

Negociar Opciones

El principio de negociar opciones surge del hecho de que bastantes hosts querrán proporcionar servicios adicionales además de aquellas dispuestas dentro de una terminal NVT convencional. Algunos usuarios pueden usar terminales sofisticadas y de ahí su deseo de trabajar con una presentación elegante en vez de la mínima establecida. Existen ciertas opciones estructuradas bajo el protocolo Telnet que pueden ser habilitadas o deshabilitadas por medio de los comandos "*DO, DON'T, WILL, WON'T*", con los cuales el usuario y el servidor acuerdan un grupo más elaborado o simplemente diferente de las convenciones particulares para su terminal durante la conexión. Tales opciones pueden incluir el cambio en la configuración de los caracteres, el modo eco, etc.

El proceso para usar las opciones es que algunas de las partes de la conexión lo solicite y la otra parte lo acepte o rechace. Cuando la opción es aceptada, inmediatamente toma efecto; si es rechazada se mantendrá lo especificado en una terminal NVT estándar.

Simetría en la sintaxis de negociación

El proceso de negociación puede conducir potencialmente hacia ciclos o bucles infinitos de reconocimientos. Alguno de los extremos que observa los comandos entrantes no como mensajes de conocimiento (ACK) sino como nuevas peticiones entonces determina que necesitan ser reconocidos, es decir anunciar al otro extremo que los ha recibido. Para evitar tales bucles, prevalecen ciertas reglas:

- Las partes solo pueden solicitar un cambio en el estado de opción; por ejemplo, un extremo no deberá enviar una petición solamente para anunciar en que modo se encuentra.
- Si una de las partes recibe lo que parece ser una petición para establecer un modo que ya esta dado, entonces dicha solicitud no debe ser reconocida (enviar un mensaje de reconocimiento). Esto con el propósito esencial de evitar ciclos infinitos en la negociación.
- Cuando uno de los extremos transmita un comando de opción hacia el otro extremo, ya sea como una petición o como mensaje de reconocimiento, y esa opción tenga un efecto sobre el proceso de los datos que se estén transmitiendo, entonces el comando será insertado en el flujo de los datos en el punto donde se desea que este haga efecto.

Es notable que pasará algún tiempo entre la transmisión de una petición y la recepción de su reconocimiento, el cual puede ser perjudicial. Por ello, es recomendable que un host mantenga datos en un buffer y los transfiera después de solicitar una opción hasta que sepa si la solicitud ha sido aceptada o rechazada, y esto para eliminar la inestabilidad del tiempo para el usuario.

5.1.1.- SECUENCIAS DE ESCAPE

Se ha señalado que un programa servidor de Telnet debe ser capaz de manejar una multiplicidad de clientes, algunos corriendo bajo el mismo sistema, otros en sistemas IBM, Macintosh, Amigas, etc. Para llevar a cabo esta función, es necesario contar con un protocolo de aplicación. Como hemos visto, Telnet opera con un protocolo de Terminal-Virtual-de-Red (NVT) donde no importa el tipo de terminal o computadora que participen en la conexión, cada una debe ser capaz de realizar correctamente operaciones tales como borrar un carácter, una línea, o hasta la pantalla completa. Además, un protocolo de aplicación regularmente permite a los programas cliente y servidor la diferenciación entre los datos enviados al usuario y los mensajes que ambos programas utilizan para comunicarse entre sí. Esto se realiza al agregar ciertos caracteres de texto en el inicio de cada línea. Por ejemplo, si el servidor envía una línea donde inicie con los caracteres TXT, entonces el resto de la línea es información para pasarse a la pantalla. En cambio, si la línea comienza con CMD, entonces se trata de un mensaje desde el software servidor al software cliente. Obviamente el usuario nunca notará estos caracteres, pues para el momento en que reciba algún dato, la información de control habrá desaparecido.

El protocolo Telnet, como muchos otros protocolos de Internet, utilizan un intercambio de comandos y respuestas para ejecutar sus operaciones en la red. En Telnet cualquier extremo de la conexión puede transmitir comandos, es decir que fluyen en ambas direcciones: cliente/servidor y servidor/cliente. Para transmitir esos comandos, Telnet se basa en caracteres conocidos como "*secuencias de escape*".

Una secuencia de escape utiliza un carácter ("*carácter de escape*") para poder identificar el inicio de un comando. El protocolo Telnet se refiere a ese carácter de escape como "IAC" (interpret as command) y su código decimal en ASCII es 255. Cada comando de Telnet debe empezar con *IAC* para poder ser identificados como tales.

La siguiente tabla menciona algunos de los comandos del protocolo Telnet identificados con la secuencia de escape *IAC*:

Nombre	*Código*	*Descripción*
EOF	236	Fin de archivo
SUSP	237	Suspender proceso actual
ABORT	238	Abortar el proceso
EOR	239	Fin de registro
SE	240	Fin de parámetros de la sub-negociación
NOP	241	Sin operación
DATA MARK o DM	242	La porción de datos de una señal SYNCH
BRK	243	Caráeter de interrupción NVT
IP	244	Proceso de Interrupción
AO	245	Abortar salida
AYT	246	Función "Are You There"
EC	247	Borrar carácter
EL	248	Borrar línea
GA	249	Señal de proseguir
SB	250	Indica que lo siguiente es una subnegociación de la opción indicada
WILL (código de opción)	251	*[Parámetro]*
WON'T (código de opción)	252	*[Parámetro]*
DO (código de opción)	253	*[Parámetro]*
DON'T (código de opción)	254	*[Parámetro]*
IAC	255	Secuencia de escape

Comandos de Telnet que identifica el carácter IAC

La aplicación basada en el protocolo Telnet es la que interactúa directamente con el protocolo a través de las secuencias de escape. Es decir, el programa de aplicación puede construirse en un lenguaje manejando las secuencias de escape. Un ejemplo de una secuencia de escape sería *IAC SUSP* que indica al receptor que interrumpa el proceso en operación.

Telnet utiliza los comandos *WILL, WON'T, DO* y *DON'T* junto con un parámetro de opción para negociar las "opciones" de la conexión Telnet (diferentes a las establecidas por NVT). La negociación de opciones requiere de tres bytes: el carácter de escape *IAC*, el byte de comando (*WILL, WON'T*, etc.), y el código de la opción. Por ejemplo, una de las partes puede enviar una petición para modificar la longitud de la línea, si es aceptada, entonces se emplean campos para negociar esa longitud. Telnet incluye más de 40 códigos negociables, los cuales pueden ser consultados en el documento RFC "Internet's Assigned Numbers" para un estudio más profundo.

El propósito de Telnet es el de proveer una interfaz estándar para las terminales y procesos orientados a terminales a través de la red. La experiencia ha mostrado que ciertas funciones son manejadas por la mayoría de los servidores, pero la manera de invocar a esas funciones difiere ampliamente. Para una persona que trabaja con varios sistemas de servidores, estas diferencias pueden ser frustrantes. Por ello, Telnet define una representación estándar para 5 de esas funciones.

IP (Proceso de Interrupción)

Muchos sistemas suministran una función para poder suspender, interrumpir, abortar, o terminar la operación de un proceso del usuario. Usualmente esta función es empleada cuando el proceso ha caído en un bucle infinito o cuando un proceso ha sido activado accidentalmente. *IP* es la representación estándar de Telnet para invocar esa función.

AO (Abortar Salida)

AO es la función que permite a un proceso que esta generando datos de salida, el poder esconder esos datos a la terminal del usuario. Por ejemplo, cuando un sistema acepta el comando de un usuario, este envía en respuesta una cadena de texto hacia la terminal del usuario , y finalmente señala la disposición para aceptar el próximo comando transmitiendo un carácter de "prompt" . Si *AO* fuera recibido durante la transmisión de la cadena de texto , la acción razonable sería omitir el resto de la cadena de texto excepto el carácter del prompt.

AYT ("Estas ahí")

Esta función permite al usuario saber si el sistema sigue funcionando. Esto puede ser requerido cuando el sistema permanece en silencio por un largo tiempo .

EC (Borrar Carácter)

Esta función borra el último carácter precedente del flujo de los datos suministrados por el usuario.

EL (Borrar Linea)

Esta función se encarga de borrar todos los datos de la línea actual en proceso.

5.1.2.- SEÑAL SYNCH DE TELNET

La mayoría de los sistemas de tiempo compartido abastecen mecanismos que permiten a un usuario de terminal el hecho de recuperar el dominio de un proceso fuera de control; las funciones *IP* y *AO* descritos anteriormente son ejemplos de estos mecanismos. En computadoras monousuario, la mayoría de los sistemas operativos reconocen ciertos tipos de interrupciones que le indican detener un programa o proceso. Por ejemplo, DOS reconoce la combinación CTRL+C, Windows reconoce la combinación CTRL+ALT+DEL. Ambas combinaciones envían una señal de interrupción al sistema operativo.

En las computadoras que forman parte de una red, los mecanismos de control de la red pueden retener una señal de interrupción. Por ejemplo, una señal de interrupción transmitida a otro host puede quedar atorada en un buffer de datos de salida debido a una congestión en el tráfico de la red. Como ya lo hemos revisado, TCP utiliza un modo de comunicación full-duplex (ambas direcciones simultáneamente). En consecuencia, mientras la señal de interrupción que ha transmitido el usuario quede paralizada en el tráfico de la red, la información del otro extremo de la carretera del sistema (información conducente al usuario) quizá siga arribando. Para combatir este problema, Telnet define un mecanismo de sincronización SYNCH. Una señal SYNCH de Telnet consiste de una notificación urgente de TCP y el comando DATA MARK de Telnet. Sabemos que la notificación "urgente" de TCP no es tópico para el control del flujo. Por ende, Telnet utiliza la notificación "urgente" de TCP para sobrepasar el control de flujo y solicitar al modulo TCP del receptor que procese inmediatamente los datos urgentes en ese canal de comunicación.

Cuando un segmento TCP arriba con la señal de bandera URG esto indica que el segmento contiene datos o información urgente. El TCP receptor debe notificarlo inmediatamente a la aplicación receptora. TCP no espera a que la aplicación procese los datos que habían llegado anteriormente, en cambio, TCP coloca a la aplicación en "modo urgente", lo que indica a la aplicación que han llegado datos que necesitan ser procesados inmediatamente. Frecuentemente, cuando una aplicación se encuentra en modo urgente, esta lee datos de la conexión TCP hasta que se alcanza el final de los datos urgentes (como lo identifica el apuntador "urgente" en el encabezado del segmento TCP). Después de que la aplicación ha leído el último byte de los datos urgentes, TCP notifica a la aplicación que ya ha procesado el último byte.

Como ya se ha mencionado, para enviar una señal SYNCH de Telnet, una aplicación debe transmitir un segmento TCP con la bandera URG (urgente) activa y el comando DATA MARK como el último (o único) byte de datos. El comando DATA MARK de Telnet es la marca de sincronización en la corriente de datos que especifica a la aplicación receptora cuando puede retomar

el proceso normal de la corriente de datos. En otras palabras, cuando el receptor esta en el modo urgente, DATA MARK señala el final del proceso urgente. En cambio, en el modo normal, el receptor no toma importancia a la señal DATA MARK. Si TCP indica el final de los datos urgentes antes de que el receptor encuentre una señal DATA MARK, la especificación de Telnet requiere que el receptor continúe manejando de forma especial (urgente) a la corriente de datos recibida hasta que se encuentre la señal DATA MARK. En conclusión, después de que TCP coloca al receptor en modo urgente, el receptor continuará leyendo y descartará la información hasta que encuentre la señal DATA MARK.

Además, Telnet permite a un receptor el unir múltiples notificaciones urgentes. Por ejemplo, suponiendo que el receptor encuentra una señal DATA MARK pero TCP indica que existen más datos urgentes (no notifica el final de los datos urgentes). Entonces, la aplicación receptora debe seguir con el manejo especial de la corriente de datos (leyendo y descartando datos) hasta que encuentre otra señal DATA MARK. En efecto, la señal SYNCH limpia el canal de comunicación (entre el transmisor y receptor) de todo dato excepto comandos exclusivos de Telnet. La especificación de Telnet describe como otros protocolos pueden utilizar la señal SYNCH para propósitos similares. Por ejemplo, Ftp define un comando de abortar ABOR diseñado para cancelar una operación de transferencia de archivos. De acuerdo con esta especificación, una aplicación que utiliza la señal SYNCH de Telnet debe desarrollar los siguientes pasos:

1.- Mandar el carácter IP de Telnet
2.- Enviar la secuencia de SYNCH de Telnet. Esto es, transmitir la señal DM como el único carácter en una operación TCP de modo urgente.
3.- Transmitir el comando de datos urgentes del otro protocolo- por ejemplo, el comando ABOR de Ftp .
4.- Emitir el equivalente a la señal DATA MARK de ese protocolo.

5.1.3.- MODO DE OPERACIÓN

Para utilizar un cliente de Telnet, el usuario deberá especificar el nombre o dirección IP del host remoto al cual se desea conectar. El programa cliente establecerá una conexión TCP para el puerto 23 (el puerto estándar reconocido para la aplicación Telnet) y después el host remoto enviara una petición de identificación con el mensaje "*login:*" al cual se debe responder con el nombre o clave de usuario. Dependiendo del servidor de Telnet que se contacte, también se deberá introducir un "*password:*". Miles de servidores de Telnet existen a través de Internet, y estos proporcionan un acceso fácil a información que va desde investigaciones científicas hasta el estado del tiempo. Como en el caso del Ftp anónimo, algunos servidores de Telnet publican un login y un password que se pueden emplear para establecer sesiones de tipo público (p.ej. gophers).[12]

12 Por ejemplo el host 132.248.10.3 (condor.dgsca), que es una máquina con el servicio de un gopher pública, cuyo login para accesarlo siempre será "info" y sin password.

La forma para establecer una sesión remota con Telnet es:
$ telnet nombre_del_host_remoto (en un equipo unix)
c:\>telnet nombre_del_host_remoto (en ambiente MS-DOS)

Notemos que no importa la plataforma en que se trabaje, el comando siempre será igual.
Veamos un ejemplo de una conexión remota a un servidor en el cual el usuario tiene un login y password propio:

$ telnet 132.248.44.120
Trying..
Connected to hp-720.aragon.unam.mx
Escape character is '^]'.
HP-UX hp-720 A.09.05 A 9000/720 (tyv0)
login:
password:
$

En este ejemplo se pidió una conexión remota con un host cuya dirección IP es 132.248.44.120 y cabe mencionar que se pudo haber indicado su nombre hp-720.aragon.unam.mx en lugar de su dirección. El programa cliente de Telnet encontró el host especificado e inicializó una sesión de terminal. Una vez que la sesión comienza, aparece el mismo diálogo e información como si se estuviera en una terminal conectada directamente a dicho servidor.

A partir de que se ha realizado la conexión a una máquina remota, y hasta que no se termine dicha sesión, todos los comandos empleados serán aquellos apropiados al sistema remoto. Es decir, como el servidor hp-720 es un sistema Unix, entonces todos los comandos estándares de Unix están disponibles. Cuando la sesión remota sea finalizada (quit), todo comando subsecuente será ejecutado por el sistema local.

Lo que realmente esta sucediendo cuando se ejecuta el comando Telnet es la participación de dos programas: el *cliente*, que corre en la computadora local de donde se hace la petición de conexión con otra máquina, y el programa *servidor*, que es el software que corre en la computadora que proporciona dicho servicio (host remoto).

El programa cliente funciona una vez que se hace la petición de Telnet y se encarga de :

- .Crear una conexión de red TCP con un servidor
- .Aceptar la entrada de datos del usuario en una manera conveniente
- .Reformatear esa entrada a un formato estándar y enviarlo al servidor
- .Aceptar los datos de salida del servidor en algún formato estándar
- .Reformatear esa salida para presentarla al usuario

El software servidor de Telnet corre en la máquina que esta desarrollando el servicio; si el programa servidor no está corriendo, entonces el servicio no está disponible. En los sistemas Unix, los programas servidores son regularmente conocidos como "daemons", tareas que corren todo el tiempo en el fondo (background) del sistema. Estos servidores o daemons se encargan de:

- .Informar al software de la red que esta listo para aceptar la conexión
- .Esperar por una solicitud en un formato estándar
- .Prestar el servicio solicitado
- .Enviar los resultados al cliente en un formato estándar
- .Regresar al modo de espera

5.1.4.- MODO DE COMANDOS

Telnet transmite todo aquel carácter que el usuario teclea hacia el host remoto, con la excepción del carácter de escape. Este carácter generalmente se denota por las teclas CTRL+] ('^]') y cuando ha sido tecleado, el programa cliente de Telnet entra a un modo especial de comandos.

También se puede llegar al modo de comandos con el simple hecho de introducir el comando Telnet sin ningún nombre o dirección de host posterior.

$ telnet [enter]
telnet

Una vez que se ha llegado al modo de comandos, se verá el indicador *telnet*>, y esto significa que Telnet está listo y en espera de algún comando. Con sólo oprimir la tecla ?, el usuario obtendrá una lista abreviada de los comandos disponibles:

telnet> ?

close	cierra la conexión actual
display	despliega los parámetros del sistema
mode	entra a un modo de línea-por-línea o un-carácter-a-la-vez
open	realiza la conexión a un host
quit	finaliza el programa Telnet
send	transmite caracteres especiales
set	establece parámetros del sistema
status	presenta la información de estado de la conexión
toggle	valida parámetros del sistema
z	suspende el programa de Telnet
?	presenta la información de ayuda

Aunque existen más comandos y también bastantes subcomandos, solo algunas son usados con mayor frecuencia:

close	Este comando se encarga de finalizar la conexión actualmente en proceso. Close automáticamente desconecta al usuario del sistema remoto; y también puede salir de Telnet si cuando se hizo la conexión se especificó el nombre del host requerido.
open name	Se encarga de establecer una conexión con la máquina especificada. El nombre o dirección de la máquina es requerido para este comando. La mayoría de los programas de Telnet solicitarán ese nombre en caso de que este no sea especificado. Es necesario cerrar cualquier conexión antes de inicializar una nueva.
set echo	Habilita o deshabilita el proceso de eco (echoing). Este proceso permite que los caracteres tecleados por el usuario aparezcan en la pantalla. Generalmente, la computadora remota es responsable de regresar estos caracteres a la terminal del usuario una vez que los ha recibido. Esto es denominado eco remoto y es considerado muy fiable por el hecho de saber que el sistema remoto ha recibido nuestros comandos correctamente. El eco local se refiere a que la computadora local (el Telnet cliente) transmita los caracteres introducidos por el usuario hacia el display de la pantalla. Debido a que el eco remoto es más fiable que el local, Telnet casi siempre comienza con el parámetro "echo" en "off". Para poder habilitarlo "on", se necesita entrar al modo de comandos y teclear el comando "set echo". Para deshabilitarlo nuevamente, únicamente hay que teclear "set echo" otra vez.
set escape char	Determina el carácter de escape de acuerdo al especificado. Para esto se debe teclear el carácter deseado oprimiendo al mismo tiempo la tecla CTRL. Se recomienda que el carácter de escape indicado sea uno que no se requiera introducir mientras se realice el proceso normal, pues esto podría provocar algún problema.
quit	Nos deja fuera del programa Telnet rápidamente.
z	Suspende temporalmente la sesión de Telnet para que el usuario pueda ejecutar comandos bajo el sistema local. La conexión y otras opciones permanecen en el mismo estado cuando la sesión es recobrada. La sesión puede recobrarse por medio de comandos normales del sistema operativo. El Sistema V de Unix nos coloca en un subshell para realizar otros comandos y para regresar a la sesión de Telnet se debe salir (exit) del shell. Estos comandos dependen del sistema operativo en uso.
Carriage Return	Sin necesidad de un comando, una línea en blanco nos regresará del modo de comandos a la conexión de Telnet remota.

Al hacer una conexión a un servidor remoto no siempre nos encontraremos con la misma petición de login a la que estamos acostumbrados y veremos la información que el administrador de ese equipo ha colocado especialmente para tal servidor. En otras palabras, cada servidor es diferente, algunos tendrán buenas interfaces con el usuario y otros tendrán algunas no muy buenas. La mayoría de los servidores solicitan al usuario el tipo de terminal para trabajar, siendo "vt100" la opción más común debido a que varios emuladores de terminal y aplicaciones de windows trabajan con ella. En su primer pantalla, los servidores siempre despliegan información útil como la forma de salir de ese equipo, la manera de obtener ayuda, las políticas de ese sistema, etc. Es recomendable observar a estas indicaciones para conocer el equipo con el que trabajamos y evitar incurrir en posibles errores.

5.1.5.- CONEXIÓN EN OTROS PUERTOS

La conexión TCP de Telnet es efectuada entre el puerto del usuario y el puerto del servidor. El servidor puede recibir múltiples peticiones de conexión en su puerto reconocido para Telnet. Debido a que las conexiones TCP son full-duplex y se identifican por un par de puertos, el servidor puede establecer diferentes conexiones involucrando su puerto de servidor y diferentes puertos de usuario.

También se puede hacer uso de un programa cliente de Telnet para establecer una conexión con un host remoto a través de un puerto en particular y diferente al estándar reconocido (puerto #23). En el capítulo 2 observamos que debido a que las computadoras de Internet proporcionan una cantidad enorme de servicios, debe haber una forma para que el software de comunicación con la red pueda identificar cual programa servidor (daemons) puede responder a una petición. Esto se efectúa asignándole un número de puerto a cada uno de esos programas. Cuando el servidor se inicializa, éste indica a través del archivo */etc/services* (en sistemas unix) que puertos están disponibles y para que servicios. De esta forma, cuando un cliente requiere de alguna aplicación en particular, debe de introducir tanto el nombre de la máquina, así como el identificador de puerto para accesar a ese servicio específico.

Para poder hacerlo, la sintaxis es
telnet máquina-remota # de-puerto

Por ejemplo, un usuario puede recuperar y revisar su correo electrónico (e-mail) utilizando un cliente Telnet para conectarse a su host remoto a través del puerto 110 que es el identificador reconocido para la aplicación de correo POP3 (visto posteriormente). Obviamente, en todos los casos, el usuario debe conocer que comandos teclear y la sintaxis que estos utilizan dentro de cada aplicación.

Ejemplo:

$ telnet hp-720 110
Trying...
Connected to hp-720.aragon.unam.mx.
Escape character is '^]'.
+OK Qualcomm Pop Server derived from UCB (version 2.1.4-R3) at hp-720 starting
user ivonne
pass [password]
+ OK ivonne has 10 message(s) (72973 octets)
list
1.
2.
quit
+OK Pop server at hp-720 signing off
Connection closed by foreign host
$

Al realizar la conexión al servidor hp-720 se indicó que no se hiciera por el puerto estándar #23 sino utilizando el puerto #110[13] (puerto reconocido para POP3). Al iniciar la sesión el usuario introduce su login a través del indicador "user" y su password con el comando "pass". Una vez que se ha entrado a la aplicación, todos los comandos disponibles son aquellos que pertenecen a la aplicación POP3, el usuario puede leer su correo y salir de el por medio de los comandos particulares de esa aplicación.

Este proceso es muy utilizado por los administradores de sistema que se percatan de algunas fallas en el proceso de ciertos servicios. De hecho, se puede hacer uso del comando Telnet para probar y entender la forma directa en que opera un protocolo cuando se realiza una aplicación.

13 Cada aplicación en particular tiene definido un número de puerto (Capítulo 2)

A continuación se presenta un ejemplo del contenido (sólo una parte) de un archivo */etc/services*, el cual contiene los nombres oficiales de los servicios y sus alias con el número de puerto que ese protocolo utiliza:

```
# @(#)$Header: services,v 1.25.193.2 93/03/22 11:35:05 ash Exp $
#
# This file associates official service names and aliases with
# the port number and protocol the services use.
#
#
# The form for each entry is:
# <official service name>  <port number/protocol name>  <aliases>
#
# See the services(4) manual page for more information.
# Note: The entries cannot be preceded by a blank space.
#
#
echo            7/tcp                           # Echo
echo            7/udp                           #
discard         9/tcp           sink null       # Discard
discard         9/udp           sink null       #
systat          11/tcp          users           # Active Users
daytime         13/tcp                          # Daytime
daytime         13/udp                          #
qotd            17/tcp          quote           # Quote of the Day
chargen         19/tcp          ttytst source   # Character Generator
chargen         19/udp          ttytst source   #
ftp-data        20/tcp                          # File Transfer Protocol (Data)
ftp             21/tcp                          # File Transfer Protocol (Control)
telnet          23/tcp                          # Virtual Terminal Protocol
smtp            25/tcp                          # Simple Mail Transfer Protocol
time            37/tcp          timeserver      # Time
time            37/udp          timeserver      #
rlp             39/udp          resource        # Resource Location Protocol
whois           43/tcp          nicname         # Who Is
domain          53/tcp          nameserver      # Domain Name Service
domain          53/udp          nameserver      #
bootps          67/udp                          # Bootstrap Protocol Server
bootpc          68/udp                          # Bootstrap Protocol Client
tftp            69/udp                          # Trivial File Transfer Protocol
finger          79/tcp                          # Finger
supdup          95/tcp                          #
hostnames       101/tcp         hostname        # NIC Host Name Server
pop             109/tcp         postoffice      # Post Office Protocol - Version 2
pop3            110/tcp         postoffice      # Post office ver. 3
portmap         111/tcp         sunrpc          # SUN Remote Procedure Call
portmap         111/udp         sunrpc          #
```

```
auth            113/tcp     authentication      # Authentication Service
sftp            115/tcp                         # Simple File Transfer Protocol
uucp-path       117/tcp                         # UUCP Path Service
nntp            119/tcp     readnews untp       # Network News Transfer Protocol
ntp             123/udp                         # Network Time Protocol
netbios_ns      137/tcp                         # NetBIOS Name Service
netbios_ns      137/udp                         #
netbios_dgm     138/tcp                         # NetBIOS Datagram Service
netbios_dgm     138/udp                         #
netbios_ssn     139/tcp                         # NetBIOS Session Service
netbios_ssn     139/udp                         #
bftp            152/tcp                         # Background File Transfer Protocol
snmp            161/udp     snmpd               # Simple Network Management Protocol
                                                Agent
snmp-trap       162/udp     trapd               # Simple Network Management Protocol
                                                Traps
bgp             179/tcp                         # Border Gateway Protocol
#
```

Nota: El archivo "services" de la red es un simple archivo en ASCII que contiene información sobre los servicios de comunicación tales como Telnet, Ftp, Mail, etcétera.
Este archivo contiene en la primer columna el nombre del protocolo de servicio, en la segunda columna define el puerto estándar asignado, y la tercer columna especifica los alias correspondientes.

5.2.- TRANSFERENCIA DE ARCHIVOS < Ftp >

Aunque, probablemente, el correo electrónico sea la aplicación de Internet más ampliamente utilizada, el Protocolo de Transferencia de Archivos (Ftp) acarrea o transporta la mayoría de la información en el Sistema. Los programas de Ftp transfieren copias de archivos desde una máquina a otra. No importa la ubicación de cada una de estas máquinas, la forma en que estén conectadas al sistema, ni tampoco el tipo de sistema operativo con el que trabajen; ambas computadoras pueden utilizar este protocolo y transferir copias de archivos con la misma estructura básica del Ftp.

La primera propuesta de FTP data de 1971 y fué desarrollada en el MIT (Massachusetts Institute of Technology) en E.U. Las aplicaciones básicas de Ftp son:

- ☑ .Proporcionar acceso a archivos almacenados en servidores centrales desde computadoras personales
- ☑ .Acceso a bases de datos públicas y
- ☑ .Distribución de información a través de la gran red

En muchos sistemas, el usuario primero debe entrar a sesión antes de poder accesar a los archivos grabados en esa máquina. Para facilitar el acceso de diversos usuarios a sus archivos, algunos sistemas proporcionan un login público denominado "anonymous" (anónimo). Este Ftp se conoce como *Ftp anónimo* y por medio de él un usuario puede entrar a sesión y transferir copias de archivos sin la necesidad de adquirir una cuenta dentro de dicho sistema. En otras palabras, Ftp anónimo permite que cualquier cantidad de usuarios de diversas partes del mundo puedan tener acceso a sus archivos de información (información de todo tipo existente en Internet) en forma gratuita.

Precisamente, Internet atrae el interés de miles de usuarios, debido en gran parte a que los habilita para poder recuperar información, programas, diversos tipos de documentos, y otros recursos. En la mayoría de los sistemas conectados a Internet, para empezar una sesión de Ftp, el usuario simplemente debe teclear el comando *ftp* en el prompt del sistema operativo. Para indicar que el usuario ha inicializado una sesión de Ftp, el host cambiará el indicador del prompt para incluir la palabra *ftp* de la siguiente manera:
ftp>

5.2.1.- SESIÓN FTP

A continuación describiremos como obtener archivos entre dos computadoras donde el usuario tiene un login establecido. De la misma forma que el comando Telnet, Ftp requiere que se especifique la máquina desde o hacia donde se desea transferir archivos. Esto se hace tecleando:
$ ftp máquina-remota

Así se inicializa el programa Ftp y se efectúa la conexión (de tipo TCP) con la máquina determinada. Una vez que se ha establecido la conexión, el usuario deberá introducir su login y su password dentro de ese sistema.

Por ejemplo:
$ ftp hp-720
Connected to hp-720.aragon.unam.mx
220 hp-720 Ftp server (Version 1.7.193.3) ready
Name : ivonne
Password required for ivonne.
Password: [password]
User ivonne logged in
ftp>

El nombre del login que el usuario introduzca determinará a qué archivos del sistema remoto tendrá acceso, de manera similar como si el usuario estuviera conectado localmente. El login y password introducidos deben ser apropiados dentro del sistema remoto.

Una vez que el sistema ha aceptado el nombre de la cuenta y su password, aparecerá el indicador *ftp* señalando al usuario que está autorizado para iniciar la transferencia de archivos.

Ahora analizaremos como realizar una conexión a un servidor que proporciona un Ftp anónimo. Sabemos que este tipo de Ftp autoriza a usuarios anónimos para que puedan accesar ciertos archivos de dominio público. Los RFC's son documentos donde son publicados los estándares, propuestas de estándares y algunas ideas generales de Internet. Estos documentos pueden ser obtenidos vía *ftp anónimo* en servidores oficiales que contienen dicho grupo de documentos actualizados y completos.

Una lista de estos servidores es:

ftp	nic.ddn.mil	login: anonymous
ftp	nis.nsf.net	login: anonymous
ftp	nisc.junc.net	login: anonymous
ftp	wuarchive.wustl.edu	login: anonymous
ftp	src.doc.ic.ac.uk	login: anonymous
ftp	nisc.sri.com	login: anonymous
ftp	nnsc.nsf.net	login: anonymous

Servidores oficiales que contienen documentos RFC de Internet

Existe un archivo nombrado como rfc-index.txt que contiene la lista de todos los documentos RFC's disponibles, especificando el título de la información y el número correspondiente a ese documento. Los documentos de texto tienen la forma de *rfcmmm.txt* donde *mmm* es el número del documento RFC buscado.

Por ejemplo, supongamos que deseamos obtener el documento RFC791 que contiene información respectiva al Protocolo IP de Internet. Para poder adquirir una copia de dicho archivo, lo primero sería contactar con alguno de los servidores de la lista anterior (cabe aclarar que existen muchos otros dentro de todo el Sistema de Internet).

$ ftp nisc.sri.com [enter]

Aquí el programa cliente de Ftp tratará de establecer una conexión con el servidor especificado. En respuesta a una solicitud de conexión, la mayoría de los servidores Ftp contestan con un mensaje que informa al usuario la manera de entrar en sesión al sistema:

Name: [anonymous]
331 Guest login ok, send "guest" as password.
Password: [guest]
230 Guest login ok, access restrictions apply.
ftp>

En un Ftp anónimo, el login público normalmente es "anonymous" y aunque en ocasiones Ftp puede aceptar cualquier cadena de caracteres como password, generalmente, si no se indica algún otro, se acostumbra introducir la dirección de correo electrónico del usuario, de manera que los administradores de ese sistema puedan comunicarse con ellos cuando lo juzguen conveniente.

Debemos notar que existe el patrón de una petición del cliente seguida siempre por una respuesta del programa servidor (un número y texto), este patrón representa el flujo de la comunicación cliente/servidor a través de toda sesión con Ftp.

Cuando se hace la conexión a un Ftp anónimo, el usuario ingresa a un lugar especial dentro del sistema de archivos. Este punto de inicio siempre es el mismo para todos los usuarios anónimos que accesan al host, y desde ahí el usuario puede moverse de directorios en forma jerárquica con el comando "cd", listar los directorios y transferir archivos.

Por ejemplo, para indicarle al servidor que cambie el directorio actual a *rfc* (donde se encuentran todos los RFC's):
ftp> cd rfc [enter] *petición del cliente*

El host servidor cambiará el directorio de trabajo a rfc. Para informarle al Ftp cliente que este comando se ha realizado exitosamente, desplegará un mensaje como:
250 CWD command succesful *respuesta del servidor*

y para enlistar la información disponible, utilizamos el comando "dir":
ftp> dir [enter] *petición del cliente*
200 PORT command succesful. *respuesta del servidor*
. . .(lista de archivos)

Para poder transferir el archivo deseado se utiliza el comando get:
ftp> **get rfc791.txt** [enter]
200 PORT command succesful
150 Opening ASCII mode connection for rfc791.txt (38517 bytes)
226 Transfer complete
38517 bytes received in 28.11 seconds (1.34 Kbytes/s)
ftp>

Los programas cliente y servidor del Ftp negociarán la transferencia de archivos y emplearan mensajes para mantener informado al usuario sobre su ejecución. Una vez que la transferencia de archivos ha sido completada, el usuario puede cerrar la conexión (TCP) por medio del comando *close*. El servidor responderá con un mensaje de "Goodbye" y en seguida terminará la sesión Ftp con el uso del comando *quit*:

ftp> close [enter]
221 Goodbye
ftp> quit [enter]

En realidad, los comandos utilizados en una sesión en Ftp, son muy similares a los que se emplean en el sistema operativo Unix. Veamos brevemente los comandos básicos en una sesión de Ftp.

5.2.2.- COMANDOS FTP

Los siguientes comandos están disponibles en la mayoría de los programas clientes de Ftp. Para obtener ayuda dentro de Ftp sobre los comandos disponibles en un sistema basta con teclear "help".

ascii Establece el modo ASCII para transferir archivos de texto.
Por ejemplo, cuando deseamos transferir archivos de texto debemos indicárselo a la máquina antes de realizar la transferencia.
ftp ascii [enter]

binary

Establece el modo Binario para la transferencia de archivos binarios (archivos ejecutables, imágenes, etc.).
ftp> binary [enter]

cd *directorio-remoto*

Cambia el directorio en la máquina remota.
Si deseáramos movernos al directorio /pub/internet dentro de la máquina remota:
ftp> cd /pub/internet [enter]

close

Finaliza la sesión con Ftp en una máquina remota y devuelve el cursor al modo de comandos de Ftp. Después de cerrar una sesión, se puede abrir una nueva conexión a través del comando open, o terminar definitivamente con el programa Ftp mediante el comando quit.
ftp> close [enter]
Goodbye
ftp>

delete *nombre-archivo*

Borra los archivos seleccionados dentro del sistema remoto .
Si deseáramos borrar un archivo de la máquina remota llamado indice.doc:
ftp> delete indice.doc [enter]

dir Enlista el contenido del directorio actual o uno especificado en el sistema remoto
ftp> dir [enter]

get *archivo-remoto*

Permite transferir la copia de un archivo desde la máquina remota hacia la máquina local.
Para obtener una copia del archivo index.txt que se encuentra en la máquina remota:
ftp> get index.txt [enter]

help *comando*

Despliega información sobre el comando determinado.
Por ejemplo, para consultar la sintaxis y uso del comando close:
ftp> help close [enter]

lcd *directorio*

Cambia el directorio actual al deseado en la máquina local.
Sin necesidad de salirnos de sesión podemos cambiar el directorio actual de nuestra máquina local, por ejemplo al directorio temp:
ftp> lcd temp [enter]

ls Presenta un listado breve del directorio en la máquina remota.
Si deseamos obtener el listado del directorio /mail:
ftp · ls /mail [enter]

mget *lista-de-archivos*
Obtiene múltiples archivos desde la máquina remota. La lista de archivos puede ser una lista de archivos separados por espacios o la selección de un grupo de archivos indicado con el comodín "*".
Si quisiéramos obtener todos los archivos que coincidan con la extensión .doc en el directorio remoto actual:
*ftp · mget *.doc* [enter]

mput *lista-de-archivos*
Envía una serie de archivos desde la máquina local hacia la máquina remota. Tiene la misma sintaxis que mget.
Para enviar una copia de todos los archivos con extensión .c de nuestra máquina local a la remota:
*ftp · mput *.c*

open *máquina-remota*
Realiza la conexión a la máquina señalada. Para abrir una nueva conexión primero es necesario cerrar la conexión actual (close).
Para conectarnos al servidor ftp.unam.mx:
ftp · open ftp.unam.mx [enter]

put *archivo-local*
Permite transferir un solo archivo a la vez desde la máquina local hacia la máquina remota.
Si solamente vamos a enviar el archivo tesis.doc desde nuestra máquina local a la remota:
ftp · put tesis.doc [enter]

pwd Indica el nombre del directorio remoto actual (ruta).
Si queremos saber en que directorio nos encontramos en la máquina remota:
ftp · pwd [enter]
/pub/internet (estamos en el directorio /pub/internet)
ftp>

quit Cierra cualquier conexión que este en proceso y termina el programa Ftp.
ftp quit [enter]
Goodbye
$ (regresamos al prompt del sistema local)

!

Utilizando el símbolo ! [enter] se pueden ejecutar comandos dentro del sistema local y regresamos al remoto con "exit". También, si el símbolo ! se escribe antes de ciertos comandos, estos serán ejecutados en el sistema local sin necesidad de cerrar la conexión. Por ejemplo :

ftp> !pwd Señala en que directorio de la máquina local me encuentro

ftp> !dir Despliega el listado del directorio actual en la máquina local

Para conocer cuáles máquinas dentro del Sistema de Internet prestan el servicio de Ftp anónimo, el usuario puede consultar herramientas de búsqueda (como el caso de ARCHIE).

5.2.3.- MODELO DE OPERACIÓN DE FTP

El Protocolo de Transferencia de archivos, similarmente a Telnet, utiliza cadenas de comandos NVT ASCII y códigos de respuesta. NVT[14] (Terminal Virtual de Red) es similar a un protocolo de red virtual, el cual esconde todas las diferencias relacionadas con características de las terminales. Como lo vimos en la especificación de Telnet, NVT puede describirse como un dispositivo imaginario que proporciona una interfaz estándar para unidades de despliegue de video o terminales. Se denomina NVT ASCII debido a que NVT utiliza el Código Estándar Americano para el Intercambio de Información (ASCII) para representar información en las transmisiones de red.

Además Ftp utiliza 2 conexiones TCP (a nivel de la capa de transporte) para realizar las operaciones de transferencia de información. Estas 2 conexiones se identifican como: ***conexión de control*** y ***conexión de información***. Cuando se inicia una sesión de Ftp el cliente se conecta al puerto 21 del TCP servidor y se crea la conexión de control de la sesión. La conexión de control es una comunicación común entre cliente y servidor (petición y respuesta). Aquí el Ftp que sirve (servidor) desarrolla una apertura pasiva en un puerto establecido y espera por las conexiones del cliente. Por su parte, el Ftp cliente contacta al servidor Ftp mediante su puerto estándar (puerto 21), y los programas negocian una conexión de tipo TCP (vista en el capítulo 2). La conexión de control se mantiene activa durante toda la transacción en Ftp. El cliente y el servidor intercambian cadenas de comandos NVT ASCII y códigos de respuesta mediante la conexión de control. Por otro lado, Ftp crea separadamente una conexión de información para cada operación de transferencia de archivos.

Cuando se solicita una transferencia de archivo, el cliente efectúa una conexión temporal al puerto 20 del TCP servidor, la cual se mantiene hasta que se haya completado la transferencia. La conexión al puerto 20 ha de reabrirse cada vez que se transmita un archivo.

---

14 NVT utiliza la codificación estándar de 7-bits en ASCII para todos los datos, incluyendo cartas, dígitos y signos de puntuación.

Resumiendo lo anterior, Ftp utiliza 2 conexiones por medio de TCP para realizar la transferencia de archivos: una conexión es para comandos y otra es para la transferencia de información.

Kris Jamsa y Ken Cope en su libro *Internet Programming*, señalan la siguiente figura como la configuración típica para las operaciones de Ftp.

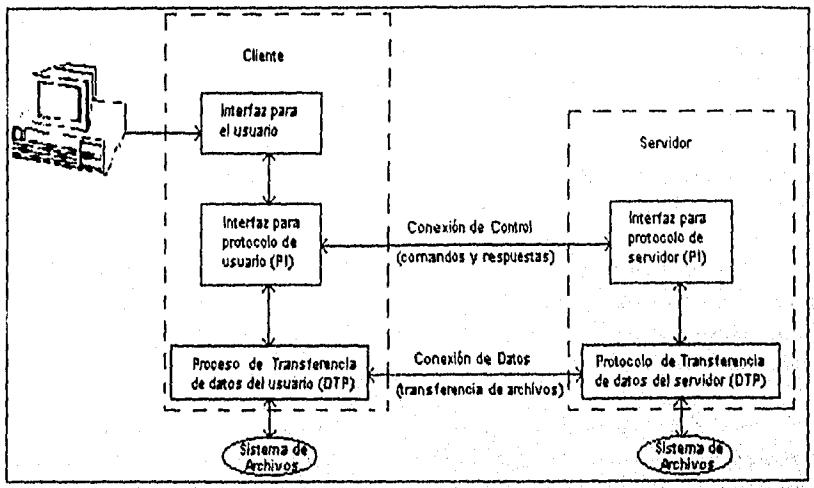


fig. 5.2 Configuración Típica para las operaciones de Ftp

La operación está a cargo de los interpretes de protocolos y de los procesos de transferencia de información . Como puede verse, cada uno de los programas cliente y servidor tienen sus propios intérprete de protocolo y proceso de transferencia de información. Los procesos de transferencia de información (DTP's) se encargan de establecer y manejar la conexión de información. Mientras que los interpretes de protocolos (PI's) traducen los comandos Ftp y los comunican a través de la conexión de control, la cual es establecida por el PI-cliente al inicio de la sesión Ftp.

La interfaz con el usuario puede ser de 2 tipos: pantalla completa (por menú) o comandos en línea. La mayoría de los programas Ftp basados en Unix se manejan por comandos en línea. Y por otro lado, los programas Ftp basados en Windows son manejados en pantallas a través de cajas de diálogos con comandos especificados en botones, cajas de listas y barras .

Manejando la información

Ftp fue diseñado para trabajar con diversas computadoras que posiblemente utilizan diferentes sistemas operativos, estructuras de archivos y grupos de caracteres. De ahí, la necesidad que el usuario pueda elegir de entre una amplia variedad de opciones para desarrollar la transferencia

de archivos. Las opciones de Ftp caen dentro de 4 categorías: tipos de archivos, formato de archivos, estructuras de archivos, y modos de transmisión.

Tipos de Archivos

Ftp puede manejar 4 tipos de archivos: local, binario (imagen o ejecutable), EBCDIC, y ASCII.

El tipo de *archivo local* soporta la transferencia de archivos entre hosts que utilizan diferentes tamaños de bytes (para algunos un byte no era forzosamente de 8 bits). Este tipo de archivo permite que un usuario transfiera datos desde un host que emplee 8 bits por byte hacia otro host que use un número diferente de bits por byte (7 o 10, por ejemplo). Para los sistemas que utilizan bytes de 8 bits (octetos), el tipo local es idéntico que el archivo de tipo binario. Como en casi todas las computadoras modernas los bytes son de 8 bits, el uso del tipo local es mínimo en las transferencias de Ftp actuales.

El tipo de *archiva imagen o binario* transfiere la información en flujo. Una transferencia de archivo imagen no identifica división alguna (como retorno del carro en el final de línea) en la estructura interna de los datos. Es muy común que los usuarios de Ftp transfieran la mayoría de sus archivos como tipo binario.

No obstante, casi todas las computadoras usan el código *ASCII* para representar datos de texto, algunos sistemas como mainframes de IBM y minicomputadoras utilizan el código EBCDIC (Código de Intercambio Decimal Codificado en Binario Extendido). Aunque ambos códigos emplean 8 bits para representar caracteres, estos son muy diferentes entre sí. Esto origina que si una computadora maneja EBCDIC no podrá entender ASCII y viceversa.

La transferencia del tipo de archivo *EBCDIC* en Ftp es un método alternativo para 2 computadoras que utilizan la codificación EBCDIC. La transferencia de archivos de tipo ASCII es el tipo establecido por default en las transferencias de Ftp. Para poder usar la transferencia de archivos en ASCII, el host transmisor debe convertir el archivo de texto local en NVT ASCII (ASCII de 7 bits). Después el host receptor traducirá NVT ASCII a la convención local para almacenar el texto.

Los clientes Ftp emplean el comando TYPE para indicar la opción deseada:

TYPE	Descripción
A	ASCII (default)
E	EBCDIC
I	Imagen o binario
L	Local

Los posibles códigos de respuesta son: 200, 421, 500, 501, 504, 530.

Formatos de archivos

Cuando un usuario elige transferir sus archivos como ASCII o EBCDIC, este también deberá especificar un control de formato. Ftp define tres tipos de control de formatos: sin-impresión, control de formato de Telnet y control de carro de Fortran.

Para los archivos de texto, el control de formato estándar es *sin-impresión*, que significa que el archivo no contiene información con un formato vertical, tales como tabulaciones verticales, que la impresora debe utilizar para colocar el texto en el papel.

El *control de formato Telnet*, utiliza controles de formatos verticales para impresoras. Los controles de formato vertical para Telnet son secuencias de caracteres que indican a una impresora como imprimir el texto.

El lenguaje de programación Fortran también emplea caracteres especiales. El *control de carro de Fortran* significa que el primer carácter de cada línea es un carácter de control de Fortran, que en turno, especificará el formato de esa línea. El control de formato de Telnet y el control de carro de Fortran son herramientas de los inicios de Internet. Actualmente la mayoría de las implementaciones de Ftp (especialmente las basadas en sistemas unix) restringen el control de formato únicamente al de sin-impresión.

TYPE	Descripción
N	con formato sin-impresión (default)
T	con formato de Telnet
C	con formato de Fortran (carriage control)

Los posibles códigos de respuesta son: 220, 421, 500, 501, 504, 530

Estructura de archivos

Ftp establece tres tipos de estructuras: archivo, registro, y página. Los Ftp clientes utilizan el comando STRU para especificar la estructura a usar en la transferencia de información. Este comando requiere solamente de un carácter como parámetro. La siguiente tabla especifica dichos parámetros:

Parámetro STRU	Descripción
F	Archivo (default)
R	Estructura de registro
P	Estructura de página

Los posibles códigos de respuesta son: 220, 421, 500, 501, 504, 530

Modos de Transmisión

El modo de transmisión determina como Ftp debe transmitir la información a través de la conexión TCP. Ftp define 3 modos de transmisión: modo de bloque, modo de compresión, y modo de flujo.

El *modo de bloque* transfiere un archivo como una serie de bloques donde cada bloque incluye uno o más bytes de encabezado. Ftp envía los bloques de información de un archivo en la misma secuencia en que se encuentran dentro del archivo. El encabezado de bloque en cada uno de los bloques de información señala el tamaño del bloque (que puede variar) para identificar el final del archivo o el final de un registro.

En el *modo de compresión*, un simple algoritmo de codificación comprime ocurrencias consecutivas de un mismo byte. Rara vez los usuarios de Ftp emplean el modo de compresión, y muchas implementaciones no le dan soporte.

En el *modo de flujo*, Ftp transfiere un archivo como un flujo de bytes de datos. Si el tipo de archivo es de "estructura de registro", Ftp utiliza una secuencia especial de 2 caracteres para marcar el final de un registro y el final del archivo. Cuando Ftp transfiere una "estructura de archivo", Ftp indica el final del archivo cerrando la conexión TCP. En otras palabras, después de que Ftp transfiere el último byte de el archivo, Ftp cierra la conexión. El receptor entiende que el cierre significa la transmisión de los últimos bytes del archivo transferido.

Los Ftp clientes utilizan el comando MODE para especificar el modo a usar. El comando requiere de un sólo carácter como parámetro como se ilustra en la siguiente tabla:

Parámetro MODE	Descripción
S	Flujo (default)
B	Bloque
C	Compresión

Los posibles códigos de respuesta son: 220, 421, 500, 501, 504, 530

Actualmente, *la mayoría de las implementaciones de Ftp solamente dan soporte a controles de formato sin-impresión, estructuras de archivos y modos de transmisión de flujo*. Esto quiere decir que, la única opción que un usuario deberá indicar es si la operación de transferencia será en ASCII o en Binario (imagen).

Es importante saber que tipo de datos son los que se desean transferir y decidir si serán transferidos como ASCII o Binario. La siguiente tabla define algunas clases de archivos y su modo de transferencia.

Archivo	*Modo*
Archivo de Texto	ASCII
Archivo de bases de datos	Binario, posiblemente ASCII
Archivo de Procesador de Texto	Binario, posiblemente ASCII
Código de Programa Fuente	ASCII
Mensajes de e-mail	ASCII
Archivo en shell de Unix	ASCII
Archivo tar de Unix	Binario
Archivo de Respaldo	Binario
Archivo Comprimido	Binario
Archivo codificado con (*uuencode*)[15]	ASCII
Archivo ejecutable	Binario
Archivo en PostScript	ASCII

Tipos de archivos comunes y su modo de transferencia

En ocasiones los archivos pueden ser muy extensos y para poder transmitirlos más rápido a través de la red, se utiliza algún método para comprimirlos. Dentro de los archivos comprimidos, existen varias técnicas para la compresión de datos, y a consecuencia un amplio número de programas diferentes para la compresión que pueden ser usados. Archivos de texto pueden ser comprimidos y reducir su tamaño de un 30% a 70 %.

Los archivos comprimidos no son un problema para transportarlos a través de la red. Estos deben tratarse siempre como archivos binarios en la transferencia. El adquirir un archivo comprimido es la mitad del trabajo a realizar, ya que después de que este es recibido, el usuario debe descomprimirlo para poder usarlo.

Los archivos bajo compresión, usualmente son identificados por un sufijo o extensión (que varía dependiendo en el programa de compresión empleado) en el nombre del archivo. Las utilerías de compresión más comunes son:

15 uuencode es una utilidad de Unix empleada para codificar archivos binarios en representaciones ASCII que facilita que sean transferidos correctamente.

Programa de Compresión	*Programa para Descompresión*	*Sufijo*	*Ejemplo*
compress	uncompress	.Z	file.txt.Z
pack	unpack	.z	file.z
Stuffit	unsit	.Sit	file.Sit
PackIt	unpit	.pit	file.pit
PKZIP	unzip41	.ZIP	file.ZIP
zoo210	zoo210	.zoo	file.zoo

El sufijo en un nombre de archivo comprimido determina que programa de descompresión deberá ser usado para obtener el archivo original.

Manejando las Conexiones

Ya hemos indicado que todos los programas clientes de Ftp utilizan la conexión de control para enviar comandos y recibir respuestas del programa servidor de Ftp. Comúnmente los comandos que envía el Ftp cliente a través de la conexión de control es para que el servidor realice alguna acción relacionada con archivos del sistema o que transfiera información por medio de la conexión de datos. Para la conexión de control, el Ftp cliente crea un socket y lo conecta al puerto estándar del Ftp servidor (puerto 21).

Para poder establecer la conexión de datos, el cliente Ftp debe seguir un procedimiento distinto. Generalmente la conexión de datos tiene 3 objetivos fundamentales:

- . Enviar una lista de directorios o archivos desde el servidor hacia el cliente
- . Mandar un archivo desde el cliente hacia el servidor
- . Transmitir un archivo desde el servidor hacia el cliente

Debido a que el cliente Ftp es quien inicia todos los comandos que requieren uso de la conexión de datos, es él quien debe crear la conexión para recibir la información solicitada, pues recordemos que utiliza la conexión de control para transmitir sus peticiones más no para la transferencia de información.

La conexión de control de Ftp permanece activa durante toda la sesión con el extremo servidor. Sin embargo, el cliente Ftp establece y mantiene la conexión de datos únicamente durante una operación de transferencia. Cada vez que el cliente requiere intercambiar información con el servidor (a través de la conexión de datos), el programa cliente crea una nueva conexión de datos. Esto implica que la conexión de datos no se efectúa sobre el mismo puerto estándar (#21), sino en otro especificado por el host del programa cliente. Si revisamos el archivo etc/services podemos observar que se especifica un puerto 20 para la conexión de datos de ftp.

De esta forma, el Ftp cliente debe desarrollar una apertura pasiva sobre el socket de la conexión de datos y después indicar al extremo servidor a que puerto del host cliente debe contactar. De otra manera, el Ftp servidor no tendrá idea de donde enviar los datos que el cliente solicitó en la conexión de control. Una vez que el cliente le ha informado al Ftp servidor cual puerto debe usar, el servidor desarrolla una apertura activa y el host del cliente emplea el socket y el puerto del protocolo que el Ftp cliente especificó. En resumen, para la conexión de datos, el Ftp cliente actúa como un servidor. El cliente crea un socket, lo liga con una dirección local, indica al servidor cual dirección debe contactar, y después espera por una conexión. Sin embargo, la diferencia entre un Ftp cliente y un programa servidor real es que un cliente solo acepta una conexión del Ftp servidor en el otro extremo de la conexión de control, mientras que el socket del servidor Ftp acepta conexiones con cualquier host remoto. Un Ftp cliente graba la dirección del Ftp servidor en el socket creado en la conexión de datos. Así, el socket aceptará solamente conexiones con el servidor Ftp. El mismo proceso para crear la conexión de datos se lleva a cabo sin importar si el cliente desea transmitir o recibir un archivo. En ambos casos (transmitir o recibir archivos), el cliente Ftp desarrolla una apertura pasiva y el Ftp servidor realiza la apertura activa.

5.2.4.- CÓDIGOS DE RESPUESTA DE FTP

En los ejemplos se muestra que el esquema de comunicación cliente/servidor consiste en que para toda petición de un Ftp cliente, el servidor siempre contesta con un código de respuesta.

El protocolo Ftp utiliza códigos de respuesta identificados por 3 dígitos (xyz). La tabla que sigue describe los códigos de control que actualmente están definidos dentro de Ftp:

Código	*Descripción*
110	Reinicia el marcador de respuesta
120	Servicio listo en nnn minutos
125	Conexión de datos ya abierta; comienzo de transferencia
150	Estado correcto de archivo; a punto de abrir conexión de datos
200	Comando correcto
202	Comando no implementado, inútil para este sitio
211	Respuesta del estado del sistema, o de ayuda del sistema
212	Estado del directorio
213	Estado de archivo
214	Mensaje de ayuda
215	Nombre del tipo del sistema
220	Servicio listo para un nuevo usuario
221	Servicio cerrando la conexión de control. Fuera de sesión si es apropiado
225	Conexión de datos abierta; sin transferencia en progreso
226	Cerrando la conexión de datos. Proceso de archivo exitoso
227	Entrar a modo pasivo

230	Procede la entrada a sesión del usuario
250	Solicitud exitosa en el proceso de un archivo; acción completada
257	PATHNAME creada
331	Nombre de usuario correcto, necesita password
332	Necesita una cuenta para accesar
350	Acción solicitada sobre un archivo esperando por mayor información
421	Servicio no disponible, cerrando la conexión de control
425	No se puede abrir la conexión de datos
426	Conexión cerrada, transferencia abortada
450	Acción solicitada para un archivo no efectuada. Archivo no disponible
451	Acción solicitada abortada: error local en proceso
452	Acción solicitada no efectuada. Espacio insuficiente de almacenamiento en el sistema
500	Error de sintaxis, comando no reconocido
501	Error de sintaxis en parámetros o argumentos
502	Comando no implementado
503	Secuencia errónea de comandos
504	Comando no implementado para tal parámetro
530	No hay acceso a sesión (not logged in)
532	Requiere de cuenta para grabar archivos
550	Acción solicitada no efectuada. Archivo no disponible
551	Acción solicitada abortada: tipo de página no reconocido
552	Acción de archivo solicitada abortada. Excede localidad de almacenamiento
553	Acción solicitada no efectuada. Nombre de archivo no permitido

5.2.5.- ABORTANDO UNA OPERACIÓN DE TRANSFERENCIA

En el mismo caso que lo especificado en Telnet, Ftp puede hacer uso de la señal SYNCH para colocar al servidor Ftp en modo urgente y que atienda inmediatamente a un comando enviado por el Ftp cliente. Los pasos son:

1.- El usuario del sistema inserta la señal IP (proceso para interrupción) de Telnet a través de la conexión de control.

2.- El usuario envía la señal SYNCH de Telnet (un segmento que contiene solamente el byte de la señal DATA MARK).

3.- El usuario coloca el comando de Ftp requerido (p.ej. ABOR) a través de la conexión de control.

4.- El interprete de protocolo del servidor (PI), después de recibir el byte del comando IP, revisa a través de la conexión de control para encontrar un comando Ftp.

Durante la operación de transferencia de archivos, Ftp utiliza la bandera URG de TCP para colocar al servidor Ftp en el modo urgente. Una vez que esto ocurre, el cliente transmite el comando de Ftp que desea sea resuelto. Cuando un servidor Ftp está en el modo urgente, este revisa en la corriente de datos de la conexión de control hasta encontrar una cadena de comando Ftp y después la responde. Si el host servidor no puede manejar 2 conexiones TCP al mismo tiempo, el servidor Ftp hará una pausa en la operación de transferencia, responderá a la petición de datos urgentes, y después proseguirá con las operaciones de transferencia.

Veamos el siguiente ejemplo:
ftp> ***mget *.exe***
mget cplay.exe ? |y|
200 PORT command succesful
150 opening Binary mode data connection for cdplay.exe (68250 bytes)
<CTRL + C>
426 Connection closed, transfer aborted
226 ABOR command successful
continue with mget ?

El usuario tiene una sesión ftp y desea transferir todos los archivos con extensión .exe dentro del directorio actual. La transferencia empieza con un archivo nombrado cdplay.exe el cual tiene 68250 bytes . Por alguna causa la transferencia del archivo esta tardando, o el usuario simplemente no desea seguir con la operación, y opta por abortar la transferencia oprimiendo las teclas CTRL + C. El sistema sabe que dicho comando es el indicado como ABOR para Ftp y cierra la conexión para cancelar la transferencia de ese archivo. Nótese que Ftp permite al usuario abortar la transferencia del archivo actual y aún puede continuar transfiriendo los archivos restantes que tengan extensión .exe si así lo requiere.

Finalmente, podemos hacer énfasis en que el acceso a todo tipo de información sin tener que pagar por la adquisición de una copia, es realmente una de las mayores atracciones de Internet. A cambio, lo que se espera de los usuarios que viajan a través de esta Carretera de Información es que se haga un uso responsable de estos recursos, tomando cada quién el sendero que le convenga, viajando libremente, pero siempre conscientes de realizar su travesía con la formalidad requerida, evitando acciones y percances que puedan accidentar al Sistema.

5.3.- CORREO ELECTRÓNICO
< E-Mail >

El correo electrónico no fue considerado como una aplicación muy relevante en el desarrollo inicial de ARPANET, sin embargo, rápidamente se convirtió en la aplicación más importante.

En la mayoría de las redes de Internet, el correo electrónico (e-mail) es la aplicación más conocida y mayormente empleada. En efecto, de todas las conexiones TCP que los usuarios de Internet establecen, aproximadamente la mitad son para el envío y recepción de mensajes por e-mail.

El tiempo de entrega para el e-mail consiste de dos partes: el tiempo que le toma a la red el entregar el mensaje a la computadora de destino (mailbox) y el tiempo que dilate el usuario en leerlo una vez que ya lo ha recibido. El primer punto esta en función de la manera en que está enlazada la computadora a la red; que puede ser mejorado con la inversión de dinero. La segunda parte esta bajo el control del usuario. El correo electrónico se considera más eficaz mientras se reduzca el tiempo de entrega entre máquina y usuario.

Los elementos básicos que todo sistema de e-mail debe incluir en una red por cada mensaje de correo electrónico son: un transmisor y un receptor, donde ambos incluyen una interfaz para el usuario (aplicación para correo) dentro del sistema e-mail de la red.

El sistema de e-mail de una red consiste de una hilera de mensajes salientes (colección de mensajes), un proceso cliente, un proceso servidor, y buzones o recipientes de correo (mailboxes) para los mensajes entrantes. Aunque con frecuencia la interfaz para el usuario (el programa de aplicación de correo) es una parte integral del sistema de e-mail, no necesariamente debe serlo. En otras palabras, la interfaz para el usuario puede ser un programa cliente por separado y que emplea un modelo cliente/servidor para interactuar con el sistema de e-mail. Un buzón de correo (mailbox) puede referirse a la dirección de un usuario (identificación del usuario y nombre del host - *login@nombre.dominio*) o a un archivo contenedor que almacena mensajes de correo.

El sistema de correo electrónico de Internet incorpora los mismos conceptos establecidos en la sección anterior. Sin embargo, la figura posterior señala los componentes reales que el sistema de e-mail de Internet utiliza .

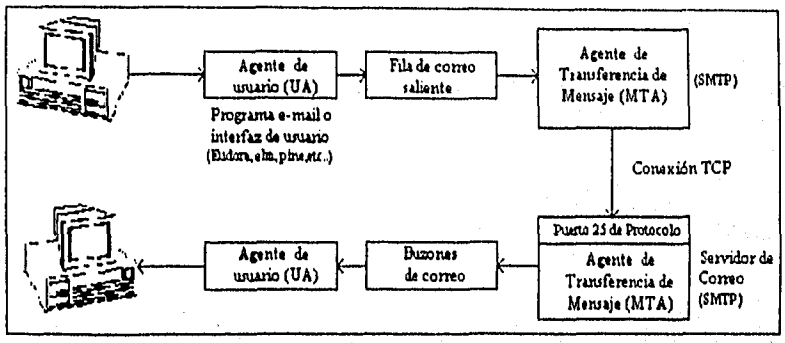


fig. 5.3 Los componentes esenciales del sistema de e-mail de Internet

Nótese el uso de términos como *agente de usuario* (UA) y *agente de transferencia de mensaje* (MTA). El "agente de usuario" se refiere al programa e-mail y el "agente de transferencia de mensaje" a los procesos cliente y servidor del proceso.

Generalmente la documentación de Internet utiliza el termino agente para referirse a un software de propósito especial que desarrolla un servicio para una persona o para otro programa. La mayoría de las especificaciones de Internet se refieren a un programa de e-mail como un agente de usuario (UA). De igual forma, un agente de transferencia de mensaje (MTA) es un programa cliente o servidor que desarrolla servicios relacionados con el e-mail, tales como enviar o recibir correo para una computadora host.

El usuario interactúa con un programa de agente de usuario, el cual, en su turno, interactúa con un contenedor de correo electrónico (o posiblemente con un programa MTA). El agente de usuario nos evita como usuarios el hecho de interactuar con una gran variedad de computadoras diferentes de e-mail. Igualmente, el MTA protege a las computadoras de una amplia variedad de agentes de usuarios u otros MTAs.

Conceptualmente, el agente de usuario (interfaz para el usuario) de un sistema de e-mail es algo separado del agente de transferencia de mensaje. Aunque se pueden colocar tanto el agente de usuario como el agente de transferencia de mensaje en un solo programa, es aconsejable apartar el diseño de cada agente en módulos separados. Aunque ambos agentes están relacionados, estos desarrollan funciones muy diferentes. Muchos de los usuarios de Internet están familiarizados con programas de e-mail tales como: MH, Berkeley Mail, Elm, Mush, y Pine (en plataforma Unix) . Y aquellos que emplean Windows como el programa de e-mail PC-Eudora y Microsoft Exchange. Cada uno de esos programas es un agente de usuario. Cada uno proporciona a los usuarios una interfaz con el sistema de e-mail de Internet. El propósito de un programa de e-mail (agente de usuario) es el de presentar un sistema de correo electrónico sencillo y amigable.

En el sistema de Internet, los agentes de transferencia de mensajes (programas clientes y servidores) representan al sistema de correo electrónico de la gran red. Los agentes de transferencia de mensajes (MTAs) que establecen conexiones de tipo TCP para comunicarse con otros MTAs asiduamente utilizan el Protocolo de Transferencia de Correo Simple (SMTP).

5.3.1.- PROTOCOLO DE TRANSFERENCIA DE CORREO SIMPLE << SMTP >>

El núcleo del sistema de e-mail de Internet son los agentes de transferencia de mensajes, pues son ellos los que se encargan de transferir el correo entre las computadoras del sistema. El Protocolo de Transferencia de Correo Simple (SMTP) representa al agente de transferencia de Internet y su propósito es el de realizar una transferencia de correo en forma fiable y eficiente. SMTP es independiente del subsistema de transmisión en particular y solamente requiere un canal que maneje la corriente de datos en forma ordenada. Una característica importante de SMTP es su capacidad para enviar correo a través de ambientes de servicio de transporte. Estos servicios de transporte proporcionan ambientes de comunicación para procesos internos. Los ambientes de comunicación pueden cubrir una red, o un grupo dentro de una red. El correo es una aplicación basada en la comunicación de interprocesos, y se comunica entre hosts de diferentes servicios de transporte mediante un host intermediario que es conocido por ambos sistemas de transporte.

5.3.1.1.- MODELO DE SMTP

Cuando un usuario realiza una solicitud de correo, el SMTP transmisor establece una comunicación de dos vías entre los MTAs del cliente (local) y del servidor (remoto). El MTA cliente envía comandos al MTA servidor, el cual, a su vez, manda respuestas al MTA cliente. En resumen, los comandos del SMTP transmisor requieren respuestas del módulo SMTP receptor. El protocolo SMTP se refiere a este hecho de intercambiar comandos SMTP y respuestas entre los dos hosts (MTAs) como una transacción de correo.

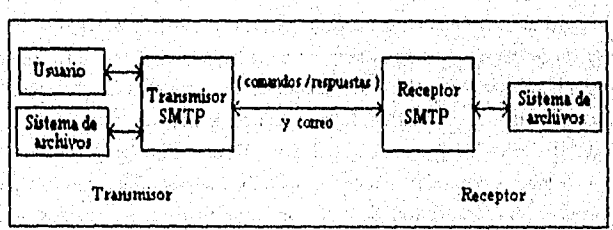


fig. 5.4 Modelo del uso de SMTP

SMTP, de forma similar a Telnet y Ftp, se basa en NVT para definir ciertas notaciones que identifican a los comandos enviados al receptor para desarrollar las operaciones de transferencia de correo. La tabla siguiente provee una breve descripción de los comandos de SMTP :

Comando	*Requerido*	*Descripción*
HELO	X	Introduce al SMTP transmisor con el SMTP receptor
MAIL	X	Indica una transacción de correo que eventualmente transfiere mensajes de correo a uno o más buzones de correo. Especifica quien es el transmisor.
RCPT	X	Identifica un recipiente individual para los datos de correo.
DATA	X	El SMTP receptor trata a las líneas que siguen al comando DATA como datos de correo. Para SMTP, la cadena de caracteres <CRLF>.<CRLF> identifica el final de los datos de correo.
RSET	X	Resetea el sistema.
NOOP	X	Requiere que el SMTP receptor no desarrolle acción alguna que la de regresar una respuesta de OK. (usado para propósitos de prueba de cliente/servidor)
QUIT	X	Requiere que el SMTP receptor regrese una respuesta de OK y cierre el canal de transmisión.
VRFY		Pide al SMTP receptor que confirme o verifique que el argumento identifica a un usuario.
SEND		Inicializa una transacción de correo que entrega datos a una o más terminales. SEND entrega datos a terminales en lugar de buzones de correo.
SOML		Inicializa una transacción SEND o de correo que entrega datos de correo a una o más terminales o buzones de correo.
SAML		Inicializa una transacción SEND y de correo que entrega datos de correo a una o más terminales y buzones de correo.
EXPN		Pide al SMTP receptor que confirme que el argumento identifica una lista de correo y, si es así, que regrese (expanda) el número de miembros de la lista.
HELP		Este comando proporciona cierta información de ayuda.
TURN		Requiere que el SMTP receptor envíe una respuesta de OK y tome el papel de SMTP transmisor o que regrese una respuesta de rechazo y mantenga el papel de SMTP receptor.

Comandos incluidos en el Protocolo SMTP

De acuerdo a la especificación, una aplicación basada en SMTP, por mínima que sea, deberá de incluir los comandos marcados como requeridos (X) en la tabla anterior. Los otros comandos de SMTP son opcionales. Cada comando SMTP termina ya sea con un espacio (si le siguen argumentos) o por un retorno de carro al inicio de línea (CRLF).

Conforme las aplicaciones de Internet han ido evolucionando, también ha incrementado el tipo de datos enviados a través de la red, aunque aquí se analizara únicamente la transacción de correo más simple (solo texto), se utiliza el termino *datos de correo* en lugar de únicamente *mensajes de correo* debido a que extensiones de SMTP[16] permiten a los MTAs el transferir archivos de imágenes, audio, y video. Esto significa que el protocolo SMTP y protocolos con extensiones de correo asociados pueden transferir más que mensajes de texto.

Tal como se indico en párrafos anteriores, SMTP provee dos vías para la comunicación entre los MTAs cliente y servidor. Los clientes mandan comandos a los servidores, y los servidores envían respuestas a los clientes. No obstante, SMTP restringe la secuencia (orden) en el que los comandos de SMTP deben ocurrir.

Una vez que se establece el canal de transmisión, el SMTP transmisor envía un comando MAIL indicando quien es el transmisor. Si el SMTP receptor puede recibir correo, entonces contestará con un mensaje OK. Después el transmisor señala a quien va dirigido el mensaje por medio del comando RCPT (recipiente). Si el SMTP receptor es capaz de aceptar correo para dicho destinatario, contesta OK; en caso contrario, responderá con una negativa para tal recipiente. Ambos extremos de la conexión SMTP pueden negociar varios destinatarios (recipientes) para un mismo mensaje. Cuando se han hecho estas negociaciones, entonces el SMTP transmisor envía los datos de correo en una secuencia determinada. Si el SMTP receptor procesa correctamente los datos de correo, este transmitirá una respuesta OK.

Para entender esto, veamos el siguiente ejemplo sobre una transacción de correo. El puerto preestablecido para el protocolo SMTP es el #25, entonces realizaremos una conexión telnet a un servidor a través del puerto 25:

$ telnet hp-720.aragon.unam.mx 25
Trying . . .
Connected to hp-720.aragon.unam.mx.
Escape character is '^]'.
220 hp-720.aragon.unam.mx HP Sendmail (1.38.193.4/16.2) ready at Tue, 25 Jun 1996 17:30:51 -0500

Cuando un MTA cliente (basado en SMTP) establece una conexión TCP para el puerto de protocolo 25, el MTA servidor (de SMTP) responde con un código de respuesta 220, el que significa que los servicios de correo de SMTP están listos.

---

16 Para mayor información consultar: RFC1869 "ESMTP", RFC1521 y RFC1522 "MIME".

Después de que los MTA establecen la conexión y el MTA servidor (receptor) envía el código de respuesta 220, SMTP requiere que el MTA cliente (fuente) transmita el comando HELO como su primer comando. Como se observa , el MTA cliente transmite el comando HELO con el nombre del host cliente como argumento. Es decir, MTA dice "Hola, soy el host hp-720".

helo hp-720

250 hp-720.aragon.unam.mx Hello hp-720(hp-720.aragon.unam.mx) pleased to meet you

El receptor (servidor) transmite un código de respuesta 250 en respuesta[17] al comando HELO. El código 250 indica al transmisor que la acción solicitada está bien y completada.

Después de que el MTA cliente establece la conexión TCP, se identifica a si mismo con el MTA servidor (usando HELO), y recibe una respuesta del servidor, el MTA cliente comienza la transacción de correo. Para iniciar la transacción , el MTA cliente transmite uno de los siguientes comandos: MAIL, SEND, SOML, o SAML. Todos los comandos MAIL, SEND, SOML, Y SAML, utilizan la siguiente sintaxis:

MAIL espacio FROM: ruta_inversa CRLF

En la siguiente línea , el MTA cliente de este ejemplo utiliza el comando MAIL para llevar a cabo la transacción.

mail from: info@hp-720.aragon.unam.mx

250 info@hp-720.aragon.unam.mx . . . Sender ok

Los argumentos de la *ruta_inversa* (info@hp-720.aragon.unam.mx) indican al MTA servidor como enviar de regreso mensajes de error al transmisor original de e-mail. Posteriormente el MTA servidor transmite otro código de respuesta 250 para indicar que la dirección de correo de la fuente es aceptable, el MTA cliente debe señalar el recipiente para los datos de correo. Para hacerlo, el MTA cliente transmite el comando RCPT (que identifica un recipiente individual para datos de correo) una vez para un solo recipiente o varias veces para identificar recipientes múltiples para los mismos datos de correo:

rcpt to: ivonne@hp-720.aragon.unam.mx

250 ivonne@hp-720.aragon.unam.mx . . . Recipient ok

rcpt to: magda@hp-720.aragon.unam.mx

550 magda . . . User unknown

---

17 Como la transacción de correo será efectuada dentro del mismo servidor, la respuesta de saludo contiene el misma host para el SMTP cliente y el SMTP servidor.

Para cada comando RCPT, el cliente MTA espera recibir un código 250. Sin embargo, en respuesta al recipiente "magda", el MTA servidor transmite un código 550. Un código de respuesta 550 en SMTP significa que el MTA servidor no puede completar la petición del cliente -la dirección no esta disponible. El usuario magda del ejemplo, no tiene una cuenta de correo en el servidor hp-720. Para infomar al MTA cliente que el buzón de correo magda@hp-720.aragon.unam.mx no existe, el MTA servidor manda un código 550. SMTP requiere que el MTA servidor notifique al cliente cuando un buzón de correo especificado como RCPT no exista. No obstante, SMTP no determina que el MTA cliente haga algo al respecto.

En seguida de que el cliente usa el comando RCPT para señalar todos los recipientes, el cliente debe transmitir un comando DATA para informar al servidor que la transmisión subsecuente representa los datos de correo.

data
354 Enter mail, end with "." on a line by itself
Date: 27 May 96 11:25
From: info@hp-720.aragon.unam.mx
To: ivonne@hp-720.aragon.unam.mx
Subject: Información
" Este es un mensaje enviado de la cuenta info del
servidor hp-720..... "
.
250 Ok

En la línea anterior el transmisor coloca el comando DATA y el servidor responde con un código 354. Este código indica al cliente que inicie la transferencia de los datos de correo y que señale el final de los datos con una cadena <CRLF>.<CRLF> (una nueva línea con un punto). Los datos de correo incluyen el encabezado con los campos: Date, Subject, To, CC, y From.

Inmediatamente después de que el cliente recibe el código 354, este puede transmitir los datos de correo. El servidor almacena los datos en una fila de datos de correo y no envia mensaje alguno hasta que el cliente transmite la cadena de caracteres <CRLF>.<CRLF>, lo que significa el final de los datos. Cuando el MTA cliente transmite una nueva línea con un punto, el MTA servidor responde con un código 250. Este código especifica que la acción de correo solicitada fue terminada correctamente.

En cualquier momento durante una transacción de correo , el MTA cliente puede usar los comandos NOOP, HELP, EXPN, y VRFY. Estos comandos regresan información al MTA cliente.

Veamos unos ejemplos:

help
214- Commands
214- HELO MAIL RCPT DATA RSET
214- NOOP QUIT HELP VRFY EXPN
214- For more info use "HELP <topic>"
214- For local information contact postmaster at this site.
214- To report bugs in this implementation contact your HP representative
214 End of HELP info

noop
250 Ok

Vrfy y Expn

SMTP proporciona herramientas adicionales para verificar un nombre de usuario o extender una lista de correo. Esto se realiza con los comandos vrfy y expn que tienen cadenas de caracteres como argumentos. Para el comando vrfy, la cadena es un nombre de un usuario y la respuesta puede incluir el nombre completo del usuario y su buzón de correo.

vrfy ivonne
250 Ivonne Mejia Berzunza <ivonne@hp-720.aragon.unam.mx>
vrfy magda
550 magda . . . User unknown

Para el comando expn, la cadena identifica una lista de correo, y la respuesta incluye en varias lineas los nombres completos y los buzones de correo de los usuarios de ésta lista de correo.
expn lista
250- Ivonne Mejia <ivonne@hp-720.aragon.unam.mx>
250- Magdalia Damián <mdm@pumas.iingen.unam.mx>
250- Miguel Meléndez <meléndez@servidor.dgsca.unam.mx>

Para finalizar una transacción de correo, SMTP necesita que el MTA cliente transmita un comando QUIT. Después que el cliente transmite el comando QUIT y el servidor responde con el código 221. El código 221 reconoce la petición de cerrar el canal de transmisión.

quit
221 hp-720.aragon.unam.mx closing connection
Connection closed by foreign host.

5.3.1.2.- CÓDIGOS DE RESPUESTA DE SMTP

Ya se ha dicho que SMTP requiere que los MTAs servidores notifiquen con un código de respuesta por cada comando que ellos reciben de un cliente. Estas respuestas a comandos SMTP son enviadas para asegurar la sincronización de peticiones y acciones en el proceso de la transferencia de correo y para garantizar que el SMTP transmisor tenga conocimiento del estado del SMTP receptor. Cada comando de SMTP puede tener como respuesta solamente un código. Sin embargo, un solo código puede incluir algunas líneas de texto. Cada dígito dentro de un código de respuesta de SMTP tiene un significado especial. El primer dígito indica si el resultado del comando fue bueno (2), malo (5), o incompleto (3). Los dígitos restantes definen la explicación de un código. La siguiente tabla enlista los códigos de respuesta de SMTP :

Código	Descripción
211	Respuesta de la ayuda o estado del sistema
214	Mensaje de ayuda
220	<dominio> Servicio listo
221	<dominio> Servicio cerrando canal de transmisión
250	Acción solicitada de correo correcta y completada
251	Usuario no local, se direccionará a <ruta de direccionamiento>
354	Inicia entrada de correo; termina con <CRLF>.<CRLF>
421	<dominio> Servicio no disponible, cerrando canal de transmisión
450	Acción solicitada de correo no efectuada: buzón no disponible (ejemplo, buzón ocupado)
451	Acción solicitada abortada: error local en el proceso
452	Acción solicitada sin efectuar: insuficiencia en el almacenamiento del sistema
500	Error de sintaxis, comando no reconocido
501	Error de sintaxis en parámetros o argumentos
502	Comando no disponible
503	Secuencia incorrecta de comandos
504	Parámetro de comando no disponible
550	Acción solicitada no efectuada: buzón no disponible (ejemplo, buzón sin acceso, no existe)
551	Usuario no local; intentar en <ruta de direccionamiento>
552	Acción solicitada de correo abortada: capacidad de almacenamiento excedida
553	Acción solicitada sin efectuar: buzón de correo no permitido
554	Transacción fallida

Tabla 5.7 Códigos de Respuesta de SMTP

5.3.1.3.- AGENTES DE RELEVO

SMTP proporciona mecanismos para la transmisión del correo; directamente desde el host del usuario transmisor hasta el host del usuario destinatario, cuando ambas máquinas están enlazadas al mismo servicio de transporte o mediante servidores SMTP de relevo (intermediarios) cuando los hosts fuente y destino no están conectados al mismo servicio de transporte.

SMTP emplea el termino *ruta_de_transmisión* (forward-path) para distinguir entre una dirección de correo (que es absoluta) y la ruta (variable) que los datos de correo pueden seguir para alcanzar su destino. Por ejemplo, suponiendo que se desea enviar dos mensajes hacia el mismo host de destino. Como sabemos, ambos mensajes tendrán el mismo campo para la dirección del destinatario del correo. Sin embargo, estos mensajes probablemente no seguirán la misma *ruta_de_transmisión* (secuencia de enrutadores) para alcanzar el buzón de correo. Similarmente, cuando el receptor del mensaje envía una respuesta hacia el transmisor por cada mensaje, puede suceder que los dos mensajes de respuesta no sigan la misma *ruta_inversa* (reverse-path, otra secuencia de enrutadores) para llegar al transmisor. Con frecuencia, los administradores de sistemas desean que el correo electrónico siga una ruta específica para llegar al recipiente o receptor de correo determinado. Para direccionar el flujo de un mensaje de e-mail, un administrador de sistema utiliza los argumentos de la *ruta_inversa* y *ruta_de_transmisión* para señalar uno o más agentes de relevo. Un *agente de relevo* es un agente de transferencia de mensaje (MTA) configurado como un concentrador de correo. Para transmitir un mensaje, cada programa de correo (agente de usuario) transfiere los datos de correo hacia un MTA local, que en turno, transfiere los datos hacia el MTA de relevo (o agente de relevo). En otras palabras, un agente de relevo es un agente de transferencia de mensajes configurado para recibir datos de correo desde varios MTAs locales y después transferirlos a través de la red. En el siguiente ejemplo, *carlos@udg.mx* representa una dirección de correo y HOST*a*, HOST*b*, y HOST*c* representan agentes de relevo:

MAIL FROM: @HOSTa, @HOSTb, @HOSTc:carlos@udg.mx

Para simplificar la configuración de su correo electrónico, una organización puede arreglar que todas las computadoras host dentro de su red envíen su correo directamente hacia un host de relevo, que es el agente de relevo (una simple computadora) que maneja la comunicación de la red a través de Internet. Además, para prevenir la entrada ilegal de invasores (usuarios indeseables) en la red a través del sistema de e-mail, una organización puede emplear el host de relevo para esconder las demás computadoras de la red. Al restringir la entrada del correo electrónico en un solo host, un sistema puede localizar el problema. En otra palabras, un administrador de sistema necesita desplegar medidas de control de penetración por correo únicamente en el host de relevo.

SMTP transporta datos atravesando por uno o más hosts de relevo cuando el host de la fuente y del destino no están conectados al mismo servicio de transporte de correo. Para proveer la capacidad del relevo, el SMTP cliente debe incluir el nombre del último host de destino, así como el nombre del buzón de destino.

El argumento del comando MAIL es la *ruta_inversa*, que identifica quien envió el correo (incluyendo hosts de relevo). El argumento del comando RCPT es la *ruta_de_transmisión*, que identifica al recipiente (receptor) del correo deseado. La *ruta_de_transmisión* es una ruta de destino, mientras que *ruta_inversa* es una ruta de regreso. SMTP utiliza la *ruta_inversa* para regresar un mensaje hacia el transmisor cuando ocurre un error con un mensaje enviado. Conforme un mensaje atraviesa el sistema de Internet, estas rutas inversa y de transmisión cambian en cada parada.

En general, para configuraciones de relevo, el proceso de transacción de e-mail en SMTP trabaja de la siguiente manera. Antes de que un host transfiera datos de correo hacia el próximo host especificado en el encabezado "TO:", el host remueve su nombre del inicio de la *ruta_de_transmisión* y lo coloca al inicio de la *ruta_inversa*.

Cuando los datos de correo llegan a su destino final, la *ruta_de_transmisión* contiene únicamente el buzón de destino. En el documento RFC821 se incluye el siguiente ejemplo para explicar la forma en que cambia la ruta cuando agentes de relevo procesan datos de correo en SMTP. Cuando un agente *A* recibe datos de correo con los siguientes argumentos:

FROM: usuarioX@HOSTY.ARPA
TO: @HOSTA.ARPA, @HOSTB.ARPA:usuarioC'@HOSTD.ARPA

el agente de relevo *A* pasará los datos de correo hacia el host *B* con los siguientes argumentos:

FROM: @HOSTA.ARPA:usuarioX@HOSTY.ARPA
TO: @HOSTB.ARPA:usuarioC'@HOSTD.ARPA

Como podemos observar, el agente de relevo *A* (*HOSTA.ARPA*) removió su nombre del encabezado "TO:" y lo colocó en el encabezado "FROM:". El agente de relevo *B* hará lo mismo y la siguiente parada para los datos del correo será el buzón del *usuarioC'* en el *HOSTD.ARPA*.

De esta forma, los MTAs de SMTP construyen la *ruta_de_transmisión* y la *ruta_inversa* conforme el correo pasa de un MTA hacia el otro.

5.3.1.4.- COMPONENTES DE E-MAIL

Los profesionales de Internet habitualmente describen tres componentes del correo electrónico: un *sobre* que utilizan los agentes de transferencia de mensajes (MTA's) , *encabezados* que emplea el programa de e-mail (agente de usuario), y el *cuerpo* del e-mail que contiene el mensaje o datos que son para el receptor.

La información del *sobre* consiste de los detalles de entrega en los campos "FROM:" y "TO:", esta información relacionada con los comandos MAIL Y RCPT es la que e-mail necesita para entregar los datos de correo desde un host a otro.

La construcción del *encabezado* en SMTP consta de los siguientes campos: Date, From, Subject, Reply-To, cc, Comment, In-Reply-To, X-Special-Action, y Message-ID. El formato general del encabezado es un campo seguido por dos puntos (:), continuado por texto que representa el valor del campo.
Ejemplo:
Date :Tue, 25 Jun 1996 17:35:51 -0500
From :Buzon de Información <info@hp-720.aragon.unam.mx>
Return-Path:<info@hp-720.aragon.unam.mx>
Apparently To: ivonne@hp-720.aragon.unam.mx
Status:R
" Este es un mensaje enviado de la cuenta info del
servidor hp-720..... "

Como se mencionó antes, el *cuerpo* del e-mail incluye la información primordial que el usuario transmisor del mensaje desea que reciba el receptor. El cuerpo de los mensajes de e-mail de Internet utiliza el código NVT ASCII. En adición, una línea en blanco usualmente separa a los encabezados del cuerpo del mensaje del e-mail.

Durante una transacción de e-mail, el usuario coloca los datos del correo (cuerpo) dentro del software del agente de usuario (programa de e-mail). Acto seguido, el agente de usuario agrega los campos del encabezado (con el contenido que especificó el usuario) y entonces transfiere el cuerpo y el encabezado a un MTA. Por su parte, el MTA agrega la información de entrega (sobre) y transfiere el paquete completo de los datos de correo hacia otro MTA que puede ser un intermediario o el destinatario final.

5.3.2.- POST OFFICE PROTOCOL <<POP>>

Actualmente la mayoría de los sistemas para correo electrónico colocan los datos de correo dentro de recipientes o buzones conocidos como "mailboxes" los cuales están localizados en algún sistema servidor de correo electrónico.

El protocolo POP3 (POP versión 3) está diseñado para que los usuarios puedan accesar desde su PC y recuperar el correo que el servidor tiene reservado para ellos.

Cuando el agente de usuario (programa de correo) de un host cliente requiere colocar correo dentro del sistema de transporte, este realiza una conexión SMTP con su host de relevo. Ese host de relevo puede ser el servidor POP3 en el host cliente. Esto es, los programas pueden utilizar SMTP para transferir mensajes de correo a través de Internet y usar POP3 para recobrar correo de la red.

5.3.2.1.- MODO DE OPERACIÓN

Cuando un host cliente solicita el uso del protocolo POP3, este debe establecer una conexión TCP con el host servidor a través del puerto 110. Una vez que la conexión se lleva a cabo, el servidor responde con un saludo. El cliente y el servidor de POP3 también operan mediante el intercambio de comandos y respuestas hasta que la conexión es finalizada o abortada.

Los comandos del protocolo POP3 consisten de una notación formada por 3 ó 4 caracteres ASCII. Estos comandos, en ocasiones, son acompañados por uno o más argumentos que pueden constar de una longitud máxima de 40 caracteres.

Por otro lado, las respuestas del protocolo POP3 consisten de un indicador de estado y un código posiblemente seguidos por información adicional. Existen 2 indicadores de estado: correcto ("+OK") e incorrecto ("-ERR").

Una sesión de POP3 pasa por 3 estados diferentes mientras se encuentra en operación. El primer estado es el de *Autorización*, donde una vez que la conexión TCP ha sido establecida, el cliente debe identificarse con el servidor y si la identificación es correcta, entonces el servidor recupera los recursos asociados con el recipiente de correo (mailbox) para tal usuario.

Cuando el servidor abre el recipiente del usuario, el proceso cambia al estado de *Transacción*. En esta etapa el cliente solicita operaciones a POP3 como el listar sus mensajes de correo, leer un mensaje en específico, borrar un mensaje, etc.

Después que el cliente envía el comando QUIT, la sesión pasa al estado de *Actualización*, en la cual el servidor POP3 guarda los recursos dispuestos durante el estado de transacción y cierra la conexión TCP.

5.3.2.2.- COMANDOS

Los comandos básicos de POP3 son:

Comando	*Estado*	*Descripción*
User *nombre*	Autorización	Identifica al usuario que requiere recuperar su recipiente de correo.
Pass *password*	Autorización	El nombre del usuario debe corresponder con un password para poder abrir su correo.
Quit	Autorización y Actualización	Termina la conexión TCP y por ende finaliza la sesión con POP3.
Stat	Transacción	El servidor entrega el número de mensajes en el recipiente y el tamaño total de todos los mensajes. Los mensajes marcados para borrarse no son incluidos en el total.
List *[mensaje]*	Transacción	Despliega la información referente a un mensaje. Si no se especifica un mensaje, la información corresponde a todos los mensajes. Los mensajes marcados para borrarse no son listados.
Retr *mensaje*	Transacción	Recupera un mensaje del recipiente de correo para ser leído. No hace referencia a mensajes marcados a borrar.
Dele *mensaje*	Transacción	El servidor POP3 marca a un mensaje para ser eliminado cuando la sesión finalice. No se refiere a archivos marcados con anterioridad.
Noop	Transacción	No se realiza otra acción más que el servidor responda "+OK".
Rset	Transacción	Desmarca mensajes que habían sido elegidos para borrarse.

A continuación veamos un ejemplo de una conexión por el puerto 110 para leer el correo de una cuenta, los comandos tecleados por el usuario aparecen en letras negras y las respuestas del servidor de POP3 están en letras cursivas:

Stelnet hp-720.aragon.unam.mx 110
Trying . . .
Connected to hp-720.aragon.unam.mx
Escape character is '^]'
+OK QUALCOMM Pop server derived from UCB (version 2.14-R3) at hp-720 starting.

Primero se establece una conexión con el servidor de correo a través de un telnet en el puerto 110. El Servidor establece la conexión y está listo para identificar al usuario en la etapa de Autorización:

user ivonne [Autorización]
+OK Password required for ivonne
pass [password] [Autorización]
+OK ivonne has 1 message(s) (371 octets)

El usuario debe introducir los comandos user y password con la identificación correspondiente a su clave dentro de dicho servidor y después que han sido aceptados correctamente (+"OK"), entonces se pasa a la etapa de transacción:

stat
+OK 1 371

El comando *stat* regresa el número de mensajes (1), y el número total de bytes en ellos (371 bytes)

list
+OK 1 messages (371 octets)
1 371

El comando *list* sin parámetros, despliega el número de mensaje y su tamaño por cada uno de los mensajes contenidos en el buzón de correo.

list 1
+OK 1 371

Por otra parte, el comando list con un número de mensaje, despliega información únicamente sobre tal mensaje.

retr 1
+OK 371 octets
Received by hp-720.aragon.unma.mx (1.38.193.4/16.2) id AA01296; Fri, 21 Jun 1996 20:01:33 -0500
Date: Fri, 21 Jun 1996 20:01:33 -0500
From: raot@hp-720.aragon.unam.mx
Return-Path: : root@hp-720.aragon.unam.mx
Apparently-To:ivonne@hp-720.aragon.unam.mx
(cuerpo del mensaje)
blah blah blah.....

El comando *retr* recupera la información (encabezado y datos de correo) contenida en el mensaje especificado.

dele 1
+OK Message 1 has been deleted

El comando *dele* marca a un mensaje para ser borrado.

stat
+OK 0 0

Después de haber marcado a un mensaje para ser borrado, este ya no es tomado en cuenta para la información del comando *stat*.

list
+OK 0 messages (0 octets)
.

Si un mensaje ha sido marcado para borrarse, entonces tampoco aparecerá en la respuesta del comando *list*.

retr 1
-ERR Message 1 has been deleted

El comando *retr* no puede abrir mensajes que están marcados por el comando *dele*.

noop
+OK

Este comando solamente obtiene una respuesta "+OK" del servidor

rset 1
-ERR Too many arguments for the rset command

El comando *rset* desmarca todos los mensajes marcados por el comando *dele*. En este enunciado hubo un error (-ERR) porque el comando *retr* no necesita parámetros.

rset
+OK Maildrop has 1 messages (371 octets)

list
+OK 1 messnges (371 octets)
1 371
.

Cuando se desmarcan los mensajes a borrar, estos vuelven a tomarse en cuenta en los comandos de transacción .

quit
+OK Pop server at hp-720.aragon.unam.mx signing off.
Connection closed by foreign host.

Quit es el mensaje de la etapa de Actualización, y causa el cierre de la conexión TCP y de la sesión con POP3 para regresar al prompt del sistema operativo.

$

Para concluir, recordemos que los protocolos SMTP y POP3 son protocolos que operan mediante una sincronización de eventos. Es decir, después que alguno de estos protocolos establecen una conexión, el transmisor (cliente) envía un comando y espera por una respuesta del servidor. Esto implica que el proceso de comunicación avanza un paso a la vez. Las computadoras se basan en SMTP para transferir correo hacia la red y en POP3 para recuperar correo del Sistema.
Veamos como último ejemplo la transferencia de un mensaje utilizando el protocolo SMTP y la forma en que se recupera dicho mensaje a través del protocolo POP3:

1. *Enviando un mensaje desde una cuenta del servidor hp-720.aragon.unam.mx hacia un recipiente en el host servidor.dgsca.unam.mx:*

HP-720# **telnet servidor.dgsca.unam.mx 25**
Trying...
Connected to servidor.dgsca.unam.mx.
Escape character is '^]'.
220 servidor.dgsca.unam.mx Sendmail 5.0/SMI-SVR4 ready at Mon, 8 Jul 1996 17:18:55 -0600
helo
250 servidor.dgsca.unam.mx Hello (hp-720.aragon.unam.mx), pleased to meet you
mail from:info@hp-720.aragon.unam.mx
250 info,.. Sender ok
rcpt to:imb@servidor.dgsca.unam.mx
250 imb... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Date: Mon, 8 Jul 1996 17:18
From: info
To: imb
prueba de datos enviada desde info@hp-720
a imb@servidor
.
250 Ok
mail from:root
250 root... Sender ok
rcpt to:imb
250 imb... Recipient ok

data
354 Enter mail, end with "." on a line by itself
correo desde root@hp-720 a imb@servidor

.
250 Ok
noop
200 OK
quit
221 servidor.dgsca.unam.mx closing connection
Connection closed by foreign host.

2. *Lectura del correo de la cuenta imb@servidor.dgsca.unam.mx:*

HP-720# telnet servidor.dgsca.unam.mx 110
Trying...
Connected to servidor.dgsca.unam.mx.
Escape character is '^]'.
+OK servidor POP3 3.3(18) w/IMAP2 client (Comments to MRC@CAC.Washington.EDU) a
Mon, 8 Jul 1996 18:38:30 -0600 (CST)
user imb
+OK User name accepted, password please
pass [password]
+OK Mailbox open, 15 messages
list
+OK Mailbox scan listing follows
1 1264
2 528
3 693
4 1103
5 1098
6 1145
7 512
8 634
9 1378
10 4253
11 1337
12 1602
13 1134
14 388
15 411
.
retr 14
+OK 388 octets
Return-Path: <info>
Received: from (hp-720.aragon.unam.mx) by servidor.dgsca.unam.mx (5.0/SMI-SVR4)
id AA23129; Mon, 8 Jul 1996 17:19:29 +0600
Date: Mon, 8 Jul 1996 17:18:53 +0600
From: info
Message-Id: <9607082319.AA23129@servidor.dgsca.unam.mx>
Apparently-To: imb
Content-Type: text
Content-Length: 57
Status: O

prueba de datos enviada desde info/a@hp-720
a inb@servidor

.
retr 15
+OK 411 octets
Return-Path: <raat>
Received: from hp-720.aragon.unam.mx by servidor.dgsca.unam.mx (5.0/SMI-SVR4)
id AB23129; Mon, 8 Jul 1996 17:21:48 +0600
Date: Mon, 8 Jul 1996 17:21:48 +0600
From: root (0000-Admin(0000))
Message-Id: <9607082321.AB23129@servidor.dgsca.unam.mx>
Apparently-To: inb
Content-Type: text
Content-Length: 40
Status: O

correo desde root@hp-720 a inb@servidor

.
stat
+OK 15 17480
dele 1
+OK Message deleted
stat
+OK 14 16216
rset
+OK Reset state
stat
+OK 15 17480
quit
+OK Sayonara
Connection closed by foreign host..

5.3.3.- ALGUNOS TIPS EN LA ESCRITURA DE CORREO ELECTRÓNICO

El correo electrónico es un medio de comunicación que en ocasiones resulta un poco informal e inseguro, y no es tan fiable como el correo postal o el teléfono. Pueden existir errores durante la transmisión de un mensaje donde este puede ser regresado y la computadora sin saber que hacer, lo entregue al administrador del sistema. Algunos usuarios tratan de encriptar sus mensajes para combatir estas deficiencias de seguridad. Como regla general, no se debe confiar 100% en la seguridad del e-mail y de ahí que no es aconsejable para la transmisión de información confidencial o privada.

Algunos tips para la escritura del e-mail son:

- Nunca declarar información por e-mail que no se desee del conocimiento público. Nunca se sabe quien terminará leyendo nuestro correo.
- No enviar mensajes abusivos, hostigosos o intolerantes. Si el receptor se queja, el administrador del sistema puede ser informado de esto y ocasionar que se tomen ciertas medidas hasta que el usuario termine con ese hábito.
- Ser cuidadosos con el uso de notas sarcásticas. El receptor no está viendo nuestros movimientos ni nuestros gestos y no podrá saber si estamos bromeando y puede resultarle molesto.
- Algunos símbolos han sido definidos como auxiliares en la denotación de ciertos gestos humanos. Por ejemplo, una cara sonriente después de decir hola:
  Hola! :-)
  Por medio de tales símbolos, llamados "*emoticones*", podemos ser más expresivos en nuestros mensajes. Estos símbolos pueden ser consultados en el gopher de la UNAM (condor.dgsca).

Para facilitar la lectura de nuestros mensajes:

- Hay que mantener nuestras líneas de un tamaño razonable (menos de 60 caracteres) si deseamos que sean legibles en cualquier tipo de terminal.
- Utilizar letras minúsculas y cuando se requiera resaltar una nota recurrir a letras mayúsculas.
- No emplear formatos fuera del normal (letra bold, itálica, etc.), pues esto frecuentemente ocasiona la aparición de caracteres incomprensibles en algunos tipos de terminales.
- Leer los mensajes antes de enviarlos. Una vez que se mandan ya no puede evitarse que su destinatario los reciba.

Conclusiones

A través de la información recopilada en la redacción de este trabajo, se expusieron algunos de los principios básicos, así como el funcionamiento de los elementos primordiales que forman parte de las operaciones generales de la Super Carretera de Información.

Aunque este gran sistema de comunicación inició su evolución en Estados Unidos desde hace más de 20 años, en algunos países apenas se está desarrollando como un gran fenómeno que permite que sus usuarios pertenezcan a una nueva sociedad cuyo anhelo principal es el de pertenecer a un mismo espacio sin límites de expresión.

Se dice que una verdadera "Super Carretera de Información" debe contar con un ancho de banda suficiente para manejar todo tipo de información en tiempo real y con suma calidad, así como proporcionar un formato consistente para gráficos y video. Aunque mucho se discute sobre si Internet aún no puede ser considerado como una Super Carretera de Información debido a que la amplitud del ancho de banda en la mayor parte de la infraestructura todavía es muy limitada y por ello no le permite transmitir cierta información como video en tiempo real y con una excelente calidad, el hecho es que el término sigue siendo empleado como referencia paralela al de Internet.

En la página con dirección URL* :http://journey.cem.itesm.mx/newtalk3_5.html se menciona que "*Debido a su limitado ancho de banda; el cual es causado por la variedad de tecnologías que usa (inclusive tiene enlaces telefónicos convencionales). Internet es un camino de terracería, y no una super carretera.*"

Alejemos nuestro pensamiento en torno a esta discusión y mantengamos la vista en que si bien no tiene un ancho de banda suficiente, esta infraestructura se esta mejorando y en muchos países este sistema ya se encuentra en la etapa de planeación activa y en muchos otros en proceso de construcción. Lo cierto es que día a día se realizan millones de conexiones las cuales generan transferencias de paquetes de información entre máquinas enlazadas a través de caminos con un sin fin de rutas para alcanzar el mismo destino.

Esta gran red de redes denomina a un sistema de comunicación a nivel mundial entre un número creciente de computadoras enlazadas a esta red y que hablan un lenguaje común. Para poder establecer la comunicación entre dos computadoras, no existen fronteras ni discriminaciones de cualquier tipo. Es una carretera pública internacional que nos permite acercarnos a personas por más lejos que se encuentren geográficamente de nosotros, donde hasta ahora no existe restricción sobre la información que se intercambie y podemos viajar de un lugar a otro en cuestión de segundos.

---

* *URL significa Uniform Resource Locator e indica una dirección de Internet donde podemos accesar a un servicio. Por ejemplo: http es el protocolo usado para desplegar páginas en el WWW.*

Se deduce que Internet ha surgido como evolución de un proyecto que en 1969 el departamento de Defensa DARPA en Estados Unidos estableció en una red militar con 4 computadoras. Esta red fue nombrada DARPAnet.

Tiempo después el nombre fue cambiado a ARPAnet y para 1972 esta red ya contaba con 37 computadoras. La red creció y para inicio de los 80 ya existían aproximadamente 200 centros conectados a ella.

En 1983 ARPAnet se divide en 2 redes: la red exclusivamente militar cambió a MILNET y ARPANET para la red pública más pequeña, el término de Internet se utilizó para denominar a el conjunto de los dos sistemas.

En 1984 nace NSFNET (la red de la Agencia Nacional para la Ciencia) que interconectaba 5 centros de cómputo con acceso a instituciones educativas.

La tecnología evolucionó al grado de permitir que para 1990 existieran equipos más sencillos con alta capacidad de desempeño. De esta forma todo tipo de persona, organización o empresa podía accesar y pertenecer a la red.

Actualmente la información disponible en la red comprende áreas de todo tipo, desde investigaciones científicas, historia, educación, etc., hasta cuestiones relativas al entretenimiento.

En efecto, este sistema de comunicación mayormente difundido como Internet, no solo abarca a los dispositivos y elementos físicos de las redes que la componen, sino también a aquellos que la diseñan y estructuran, y por supuesto, a los millones de usuarios o viajeros de esta enorme fuente de información.

Internet también es definida como una red de "conmutación de paquetes" ya que los programas dividen sus mensajes en paquetes para transmitirlos a través de la red. La dirección de la máquina de destino de los paquetes es especificado en cada uno de los fragmentos mediante el uso de encabezados. Estos encabezados contienen información necesaria para que los paquetes lleguen a su destino y puedan ser reordenados para formar el mensaje original.

Una de las características que le ha permitido la incorporación de miles de redes, es el hecho de que este sistema puede enlazar redes sin necesidad de una topología o tecnología exclusiva, es decir, pueden conectarse redes con topología de estrella, bus, o anillo y de tecnologías como Ethernet, ARCnet, Token Ring, ATM; sin que esto interfiera en el proceso de comunicación con las demás redes del sistema.

Para hacer factible la comunicación entre cualquier marca o tipo de computadora, y entre cualquier tecnología y topología de red a través de Internet, se requiere del empleo de la serie de protocolos TCP/IP.

La serie de protocolos TCP/IP es el software necesario para que cada computadora pueda conectarse y establecer la comunicación con cualquier nodo del sistema. Estos protocolos presentan las reglas que deben seguirse para controlar el flujo de información entre las computadoras a través de la red.

La serie TCP/IP está conformada por los protocolos IP, TCP, UDP e ICMP principalmente. Estos protocolos se organizan en diferentes niveles (pila de protocolos) en correspondencia con el modelo OSI de una red: *capa de aplicación, capa de presentación, capa de sesión, capa de transporte, capa de red, capa de enlace, y capa física.*

El Protocolo IP (Internet Protocol) es el sistema de entrega de Internet y para enviar la información de una computadora a otra a través de la red, utiliza paquetes llamados "datagramas". Los datagramas contienen la información del usuario más un encabezado creado por la capa del protocolo IP. Este encabezado incluye los datos especiales que son requeridos para realizar la entrega de dicho paquete. Este protocolo pertenece a la capa de red de la pila de protocolos.

Los protocolos de transporte en Internet son 2: TCP (Transport Control Protocol) y UDP (User Datagram Protocol). Cuando un usuario envía datos por medio de una aplicación, este requerirá hacer uso de uno de los dos protocolos de transporte. Los protocolos de transporte son los que se encargan de "transportar" la información entre una aplicación transmisora y otra receptora. La diferencia entre ambos protocolos consiste en la complejidad de su operación y la fiabilidad del transporte de la información. TCP es mucho más complejo y fiable que UDP. El primero transporta los datos en "segmentos TCP" y el segundo en "datagramas UDP".

TCP es el protocolo que se encarga de transferir enormes cantidades de datos y verifica su estado. Usar TCP requiere demasiada complejidad y mayor tiempo, cuando los datos que serán enviados caben en un solo paquete y no existe gran importancia en garantizar la entrega, es más viable usar UDP. UDP no se preocupa por paquetes extraviados, ordenar los paquetes en secuencia, checar su estado, etc. UDP es usado por programas que sólo envían mensajes muy cortos y que pueden retransmitir el mensaje en caso de que no reciba respuesta en un tiempo determinado. En caso de que los datos se extravíen, eso dependerá de la aplicación y no de UDP.

Las aplicaciones son otra capa de software localizada en la parte superior de los servicios TCP o UDP. Las aplicaciones son una interfaz que permite al usuario realizar sus tareas de forma amigable.

El protocolo ICMP (Internet Control Message Protocol) se encarga de transmitir mensajes de control y reportes sobre condiciones de la red que necesitan ser atendidos por el software de la misma.

Lo que realmente sucede en un proceso de transmisión de información desde una computadora fuente hacia una de destino, es el encapsulamiento de los datos en cada nivel de los protocolos. Es decir, el usuario interactua directamente con una aplicación para solicitar el envío de la información. La capa de aplicación se comunica con la capa de transporte donde llama al protocolo TCP o UDP según lo requiera la petición solicitada. El protocolo de transporte elegido encapsulará esta información en el formato respectivo, "segmentos TCP" o "datagramas UDP".

La capa de transporte envía ese nuevo formato de datos a la siguiente capa que es la del protocolo IP. El protocolo IP vuelve a encapsular esa unidad de datos en un nuevo paquete llamado "datagrama IP".

La capa de red, a la que pertenece IP, se comunica con la capa de enlace donde se encuentra la tarjeta interfaz de red y dependiendo de su tecnología (Ethernet, ARCnet, Token Ring, ATM, vía modem, etc.), está reencapsulará los datagramas en arreglos de datos de dicha tecnología.

Por último, los datos pasan en bits a través de las líneas de transmisión (capa física) de la red.

Ya en la computadora de destino, la información irá pasando por estas capas en forma ascendente (desde la capa física hasta la de aplicación). Cada capa o nivel de protocolo desencapsulará el formato que le corresponde para después pasar la unidad de datos al siguiente nivel hasta que la información llegue a la aplicación de destino.

En ocasiones cuando los paquetes de datos sean demasiado grandes para viajar a través de la red, estos serán "fragmentados" (divididos) en otros paquetes para lograr la transmisión requerida. Dichos paquetes deben identificarse con un número de secuencia y a que datagrama pertenecen, pues pueden tomar diferentes rutas para llegar al mismo destino. En el host de destino esta información es útil para colocar los fragmentos en el orden adecuado y construir el datagrama original.

Todo este proceso de encapsular la información es con el objeto de direccionar los paquetes en una forma estructurada y evitar al máximo la pérdida o corrupción de los datos.

Para poder identificar a los hosts de transmisión y los hosts de destino de los paquetes, es necesario relacionar a cada una de las computadoras de Internet por medio de direcciones. Estas direcciones son asignadas a cada tarjeta interfaz de red que contenga una máquina, además las direcciones de estas tarjetas deberán ser únicas dentro de todo el sistema (no deben duplicarse). Estas

son conocidas como direcciones IP y generalmente se escriben en notación decimal, constan de 4 bytes separados entre sí por un punto.

Internet ha clasificado a estas direcciones en 4 grupos: clase A, B, C, (D y E). Esta estructuración en clases ha permitido que el sistema pueda conectar un número de 2,113,663 redes que en conjunto pueden enlazar aproximadamente 3 mil 700 millones de computadoras.

Para poder lograr la administración de la asignación de estas direcciones, existe el Centro NIC (Network Information Center) quien realiza la asignación de direcciones de red y cada red en particular deberá asignar las direcciones de las computadoras enlazadas a ella. La tarea primordial será la de asignar direcciones únicas para cada máquina de la red.

El hecho de referirse a una máquina por medio de una serie de números, algunas veces puede resultar tedioso y facilita que los usuarios olviden o confundan estos números. Por esta razón, fue creado el Sistema de Nombres de Dominio "DNS" de Internet. Este sistema permite que las computadoras tengan un nombre específico dentro de todo Internet y de esta forma los usuarios pueden olvidarse de las direcciones IP y referirse a una máquina por medio de un nombre que resulta más sencillo y fácil de memorizar.

El sistema DNS contiene una base de datos donde realiza la traducción del nombre de una máquina a su dirección IP correspondiente, la cual utilizará en el proceso de comunicación. Para controlar y administrar todos los nombres asignados, DNS está organizado en niveles jerárquicos conocidos como dominios. El nivel superior es llamado nivel de "root".

Los nombres asignados también deben ser únicos y se forman a través de etiquetas que hacen referencia al dominio al que pertenecen dentro de la estructura jerárquica de todo el sistema. La ventaja de subdividir a todo el sistema en módulos conocidos como dominios es la de permitir una mayor organización para su manejo y administración.

Para que una persona pueda ser usuario de Internet necesita obviamente enlazarse a la red a través de algún medio de comunicación. Las grandes empresas pueden adquirir un servicio dedicado mientras que los usuarios de casa lo pueden hacer mediante modems conectados a su línea telefónica más el servicio de un proveedor de Internet. En todos los casos será necesario contar con el software de comunicación TCP/IP.

Además de TCP/IP, el software requerido para establecer una conexión a Internet, dependerá del tipo de conexión al sistema.

El número más creciente de usuarios es el de aquellos que lo hacen a través de modems y con alguno de los protocolos SLIP / PPP. Estos usuarios gozan de un acceso completo a Internet.

Los protocolos SLIP y PPP son los que gobiernan la comunicación de tipo serial para poder transmitir y recibir información de un lugar a otro a través de la red. Ambos protocolos realizan el proceso de forma distinta, pero también encapsulan los datos recibidos en un nuevo formato.

Aunque SLIP es ahora más popular que PPP, no es aún un estándar documentado en Internet, mientras que PPP si es reconocido como protocolo estándar. Se espera que poco a poco PPP tomará mayor uso y predilección que SLIP.

No obstante, Internet ofrece una inmensa variedad de servicios, el correo electrónico sigue siendo la operación más conocida y efectuada entre sus usuarios. El correo electrónico o e-mail se encarga de mover mensajes a través de la red. Estos mensajes no solamente pueden contener datos de texto, como lo hemos visto aquí, también se pueden transmitir y recibir datos de audio, video e imagen por medio de otras herramientas (no incluidas en este trabajo) y protocolos del correo electrónico .

El sistema de correo de Internet consiste principalmente de "buzones de correo" (mailboxes), agentes de usuario (aplicaciones), y agentes de transferencia de mensajes (protocolos). Estos elementos hacen que la interfaz del correo electrónico con el usuario sea más amigable.

La médula espinal del e-mail de Internet recae en el protocolo SMTP (Simple Mail Transfer Protocol) para la transferencia de los mensajes de correo. El protocolo POP3, por su parte se encarga de recuperar el correo disponible en la red para un usuario determinado. Aplicaciones como PC Eudora, Microsoft Exchange, etc., hacen uso del protocolo POP3 para recuperar el recipiente de correo de una cuenta.

Otro de los servicios o herramientas de Internet más utilizados es Telnet (o sesiones remotas con otras máquinas). A través de Telnet nosotros podemos conectar nuestra máquina local a un servidor remoto de cualquier parte del mundo.

Telnet se basa en un protocolo de terminal virtual "NVT", el cual hace posible que diferentes tipos y marcas de terminales puedan ser compatibles y manejar en forma similar la información de display de video o terminal. Este protocolo NVT utiliza el código ASCII (7 bits) para especificar ciertos caracteres de display.

Además Telnet utiliza una conexión TCP entre el proceso cliente y el proceso servidor, y maneja códigos de respuesta para cada solicitud que realiza el cliente. Esto permite conocer el estado de la conexión y de los procesos en ella.

Ftp (File Transfer Protocol) es otro de los servicios descritos en este documento. Este protocolo de transferencia de archivos permite a un usuario el obtener y colocar copias de archivos entre un sistema local y uno remoto.

Este protocolo también emplea el protocolo NVT ASCII para presentar códigos de respuesta como lo hace Telnet.

Ftp emplea 2 conexiones TCP para las operaciones de transferencia de archivos: un canal para transmitir comandos e información de control y el otro canal para transportar los archivos de información.

Ftp se relaciona ampliamente con Telnet pues utiliza ciertos comandos (como SYNCH) de Telnet para poder cancelar o abortar operaciones de transferencia de archivos.

Existen servidores de "ftp anónimo" los cuales permiten el acceso a cualquier usuario para obtener archivos de dominio público. La información disponible puede ser referente a cualquier disciplina desde temas científicos, filosóficos, políticos, noticias, entretenimiento, etcétera.

La gama de posibilidades que ofrecen el Telnet, Ftp y el E-mail, entre otros, han dado lugar al enorme crecimiento de los usuarios de la red, y han propiciado el nacimiento de una sociedad informatizada.

La mayoría de los usuarios de Internet saben como usarla, y no se preocupan de como opera este sistema para poder realizar sus operaciones. Para la gente que diseña y desarrolla los programas de aplicación y servicios de la red, el conocimiento básico del sistema es esencial pues de su entendimiento dependerá la evolución de nuevos productos.

Con este trabajo pretendo presentar algunas de esas bases que impulsen estudios más detallados para la creación de nuevos proyectos y aplicaciones.

Para terminar, cabe hacer énfasis que en México como en casi todo el mundo, el término "Super Carretera de Información" está presente en cualquier medio de comunicación, en las pláticas cotidianas de investigadores, científicos, estudiantes y amas de casa. Aunque este proyecto aún no es realidad al 100%, a través de Internet podemos conocer y obtener información hasta hace unos años inconcebible.

No es necesario ser un usuario experto, ya que con el software disponible (con interfaces amigables), cualquiera puede acceder a los recursos que Internet ofrece y su avance dependerá del tiempo que le dedique.

GLOSARIO

Aplicación
Software que desarrolla cierta actividad o tarea por un usuario. Es la interfaz con la cual interactua directamente el usuario.

ARPAnet
Una red experimental puesta en marcha en 1969 de donde se desarrollaron las teorías y software en que se basa Internet .

Baudio
El número de bits que se transmiten en un segundo (baudios por segundo).

BIND
El software desarrollado para el DNS. Sus iniciales pertenecen a "Berkeley Internet Name Domain"

Buffer
Mecanismo que almacena información temporalmente

Datagrama
Un mensaje enviado en una red de conmutación de paquetes

DNS
El Sistema de Nombres de Dominio, que es una base de datos distribuida que traduce nombres de computadoras a sus correspondientes direcciones IP.

DoD
El Departamento de Defensa de los Estados Unidos

Enrutador
Un dispositivo que transfiere datos entre redes que utilizan los mismos protocolos. Las redes pueden diferir en su tecnología.

FTP
a) El protocolo de Transferencia de Archivos, el cual define como transferir copias de archivos de una computadora a otra.
b) Un programa o aplicación que mueve archivos utilizando el protocolo FTP.

Gateway
Un dispositivo que transfiere datos entre aplicaciones o redes incompatibles. Este sistema reformatea los datos de manera que puedan ser legibles para la nueva red.

Gopher
Un sistema basado en menús para explorar recursos de Internet.

IAB
"Internet Arquitecture Board" , aquella organización que realiza decisiones sobre estándares y otros temas de interés en la red.

IETF

"Internet Engineering Task Force", un grupo voluntario que investiga y soluciona problemas técnicos, y hace recomendaciones a la IAB.

Interfaz

Medio que sirve de enlace entre las partes de un sistema de transmisión

Internet

Generalmente (en minúsculas), cualquier colección de redes distintas trabajando juntas como una misma..

Especialmente (en I mayúscula), la red mundial de redes que están conectadas unas con otras y utilizan la serie de protocolos TCP/IP .

IP

Internet Protocol, el más importante de todos los protocolos de la serie TCP/IP. Este permite que paquetes de información atraviese múltiples redes para alcanzar su destino final.

ISO

"International Organization for Standardization", la organización que desarrollo una serie de protocolos de red llamados protocolos ISO/OSI.

Host

Una computadora, en particular un transmisor o receptor de mensajes desde el punto de vista de la red de comunicación.

MILNET

Una de las redes que iniciaron a Internet, la cual manejaba información exclusivamente militar de los Estados Unidos.

Modem

Un dispositivo que conecta a una computadora con una línea de transmisión de datos (una línea telefónica) para poder realizar la conexión a una red local e Internet.

NIC

"Network Information Center" , organización responsable de coordinar las funciones de una red y proporcionar información.

NOC

"Network Operation Center", grupo responsable del cuidado diario de la red , cada proveedor cuenta con un NOC.

Puerto

Un número que identifica una aplicación particular de Internet. Cuando una computadora transmite un paquete a otra computadora, ese paquete contiene información sobre el protocolo que esta usando (TCP ó UDP), y con que aplicación intenta comunicarse. El "número de puerto" identifica a la aplicación.

PPP

"Point to Point Protocol", uno de los protocolos que permite la comunicación entre una computadora e Internet mediante líneas telefónicas.

Protocolo

Los protocolos definen las reglas que deben seguirse para que 2 o más computadoras puedan establecer comunicación entre ellas. Los protocolos estándares dejan que computadoras de diferentes marcas y sistemas puedan comunicarse sin ningún problema.

RFC

"Requests for Comments" , una serie de documentos que establecen los estándares y propuestas sobre Internet.

SLIP

"Serial Line Internet Protocol" Protocolo que habilita la comunicación entre computadoras de la red a través de línea telefónica. Aún no es un estándar de Internet, pero todavía es más usado que PPP.

Socket

Una dirección que específicamente incluye un identificador de puerto, que es la concatenación de una dirección de Internet con un puerto TCP.

TCP

"Transport Control Protocol", uno de los protocolos de transporte de Internet . TCP es un protocolo orientado a la conexión, complejo y fiable.

Telnet

Un protocolo de emulación de terminal mediante el cual un usuario puede entrar a sesión en sistemas remotos de Internet.

Un programa o aplicación por el cual un usuario se conecta a un sistema remoto usando el protocolo Telnet.

UDP

"User Datagram Protocol" , otro de los protocolos de transporte de Internet. UDP es un protocolo no orientado a la conexión, muy simple y no muy fiable.

URL

"Uniform Resource Locator", indica una dirección de Internet donde podemos accesar a un servicio. Por ejemplo: si la dirección inicia con http, este protocolo es usado para desplegar páginas en el WWW.

WWW

"World Wide Web", un sistema basado en el hypertexto (ligas en documentos) para encontrar y accesar a los recursos de Internet.

BIBLIOGRAFÍA

Apuntes Colegio de Bachilleres,
Sistemas de Información y la Información en la Microcomputación,
Primera Edición,1990, Editorial Concepto S.A.,
México, D.F.

Kris Jamsa, Ph.D. and Ken Cope
Internet Programming,
Jamsa Press, 1995, U.S.A.

Krol Ed.,
The Whole Internet User's Guide & Catalog,
O'Reilly & Associates, Inc.,
September 1992, U.S.A.

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

Periódico Computer World

Editorial,
"Una Reflexión en Torno a Internet",
ComputerWorld México,
Octubre 23 - 27, 1995, pag. B10

N. Rivera,
"Queda algo por decir . . . Por un Servicio de Internet Abierto",
ComputerWorld México,
Octubre 23 - 27,1995,pag. B11

Revistas

"Internet",
Revista Soluciones Avanzadas,
Tecnologías de Información y Estrategias de Negocios,
No. 23, Julio 15, 1995

Trudy E. Bell, John A. Adam & Sue J. Lowe,
"Technology 1996, Communications",
IEEE Spectrum,
January 1996, pp. 30 - 41.

Requests For Comments

Clark David D.,
"Name, addresses, ports, and routes",
Request for Comments:814, July 1982.

Cobb S.,
"PPP Internet Protocol Control Protocol Extensions for Name Server Addresses",
Request for Comments:1877, December 1995.

Defense Advanced Research Projects Agency and Information Sciences Institute,
"Internet Protocol, DARPA Internet Program Protocol Specification",
Request for Comments:791, September 1981.

Klensin J., N. Fred, M. Rose, E. Stefferud, D. Crocker
" SMTP Service Extensions",
Request for Commentes:1869, November 1995.

Krol Ed. & E. Hoffman,
"Network Working Group",
Request for Comments:1462, May 1993.

Mockapetris P.,
"DNS encoding of Network Names and other Types",
Request for Comments:1101, April 1989.

Myers J., Carnegie Mellon & M. Rose,
"Post Office Protocol - Version 3",
Request for Comments:1725, November 1994.

Postel Jon,
"Transmission Control Protocol",
Request for Comments:761, January 1980.

Postel Jon,
"User Datagram Protocol",
Request for Comments:768, August 1980.

Postel Jon,
"Internet Control Message Protocol",
Request for Comments:792, September 1981.

Postel Jonathan B.,
"Simple Mail Transfer Protocol",
Request for Comments:821, August 1982.

Postel J. & J. Reynolds,
"Telnet Protocol Specification",
Request for Comments:854, May 1983.

Reynolds,
"File Transfer Protocol (FTP)"
Request for Comments:959, 1985.

Romano S., Stahl & Recker,
"Internet Numbers",
Request for Comments:1117, August 1989.

Romkey J.,
"NonStandard for Transmission of IP Datagrams over Serial Lines: SLIP",
Request for Comments:1055, June 1988.

Simpson W., Daydreamer,
"ICMP Domain Name Messages",
Request for Comments:1788, April 1995.

Direcciones URL

gopher://condor.dgsca.unam.mx
"Políticas de Uso Aceptable para RedUNAM",
"Servicios de Conectividad para datos en la UNAM",
"Instituciones Directamente conectadas a la UNAM",
"Instituciones Anunciadas para Internet a través de RedUNAM",
"Requerimientos para accesar a la red vía Teléfono",
"Acceso a la red vía modem para emulación de Terminal",
"Acceso a la red vía modem empleando SLIP",
"FTP (File Transfer Protocol)",
"NOC (Network Operation Center)",
"Centro de Información de RedUNAM (NIC-UNAM)",
"Parqué usar DNS ?"

http://journey.ccm.itesm.mx/newtalk3_5.htm

http://www.issi.com:80/misc/articles/scenes.html
Ted Lewis, *"Scenes From Life on the Data Superhigway"*,
Publication of the IEEE Computer Society, April 1994.