Página principal

Tutorial y descripción técnica de TCP/IP

Internet: Pasado, presente y futuro

Architectura y protocolos

Página principal

Cursillos y tutoriales

- PORTADA
- PRESENTACIÓN
- GOBIERNO DE LA UNIVERSIDAD
- SERVICIOS
- CENTROS Y DEPARTAMENTOS
- BIBLIOTECA UNIVERSITARIA
- ESTUDIANTES
- DOCUMENTACIÓN
- AGENDA Y NOTICIAS
- OTROS RECURSOS
- NOVEDADES
- DIRECTORIO Y BÚSQUEDAS

- Guía de internet
- Manual de usuario del correo electrónico (WinPMail)
- Instrucciones para configurar su PC para acceder a Internet mediante modem

- Curso de iniciación a la publicación en Web
- Tutorial de lenguaje HTML
- Otro tutorial de HTML
- Breve referencia de HTML
- Tutorial de DHTML

- Tutorial de programación HTML y CGI-BIN
- Tutorial de programación CGI
- Tutorial de Java
- Tutorial de JavaScript
- Tutorial de Perl
- Tutorial de php
- Tutorial de WML
- Tutorial de XML

- Tutorial de algorítmica y lenguaje C
- Tutorial y descripción técnica de TCP/IP

Tabla de contenidos Quinta edición

Capítulo 1. Internet: Pasado, presente y futuro

En este capítulo describiremos cómo se formó Internet, cómo se desarrolló y cómo es probable que evolucione en el futuro.

Tabla de contenidos Introducción

1.1 Introducción

Las redes se han convertido en una parte fundamental, si no la más importante, de los actuales sistemas de información. Constituyen el pilar en el uso compartido de la información en empresas así como en grupos gubernamentales y científicos. Esta información puede adoptar distintas formas, sea como documentos, datos a ser procesados por otro ordenador, ficheros enviados a colegas, e incluso formas más exóticas de datos.

La mayoría de estas redes se instalaron a finales de los años 60 y 70, cuando el diseño de redes se consideraba como la piedra filosofal de la investigación informática y la tecnología punta. Dio lugar a numerosos modelos de redes como la tecnología de conmutación de paquetes, redes de área local con detección de colisión, redes jerárquicas en empresas, y muchas otras de elevada calidad.

Desde comienzos de los '70, otro aspecto de la tecnología de redes cobró importancia: el modelo de pila de protocolo, que permite la interoperabilidad entre aplicaciones. Toda una gama de arquitecturas fue propuesta e implementada por diversos equipos de investigación y fabricantes de ordenadores.

El resultado de todos estos conocimientos tan prácticos es que hoy en día cualquier grupo de usuarios puede hallar una red física y una arquitectura adecuada a sus necesidades específicas, desde líneas asíncronas de bajo coste, sin otro método de recuperación de errores que una función de paridad bit a bit, pasando por funciones completas de redes de área extensa(pública o privada) con protocolos fiables como redes públicas de conmutación de paquetes o redes privadas SNA, hasta las redes de área local, de alta velocidad pero distancia limitada.

El lado negativo de esta explosión de la información es la penosa situación que se produce cuando un grupo de usuarios desea extender su sistema informático a otro grupo de usuarios, que resulta que tiene una tecnología y unos protocolos de red diferentes. En consecuencia, aunque pudieran ponerse de acuerdo en el tipo de tecnología de red para conectar físicamente sus instalaciones, las aplicaciones(como por ejemplo sistemas de correo) serían aún incapaces de comunicarse entre sí debido a los diferentes protocolos.

Se tomó conciencia de esta situación bastante temprano(a comienzo de los '70), gracias a un grupo de investigadores en los Estados Unidos, que fueron artífices de un nuevo paradigma: *la interconexión de redes*. Otras organizaciones oficiales se implicaron en la interconexión de redes, tales como ITU-T e ISO. Todas trataban de definir un conjunto de protocolos, distribuidos en un conjunto bien definido de capas, de modo que las aplicaciones pudieran comunicarse entre sí, con independencia de la tecnología de red subyacente y del sistema operativo sobre el que se ejecutaba cada aplicación.

1.2 Interredes

Los diseñadores originales de la *pila de protocolos ARPANET*, subvencionados por DARPA ("*Defense Advanced Research Projects Agency*")introdujeron conceptos fundamentales tales como la *estructura de capas* y el de *virtualidad* en el mundo de las redes, bastante antes de que ISO se interesase en las redes.

El organismo oficial de esos investigadores fue el grupo de trabajo en red ("Network Working Group") llamado ARPANET, que tuvo su última reunión general en octubre de 1071. DARPA ha continuado su investigación en busca de una pila de protocolos de red, desde el protocolo host-a-host *NCP("Network Control Program")* a la pila de protocolos TCP/IP, que adoptó la forma que tiene en la actualidad alrededor de 1978. En esa época, DARPA era un organismo famoso por ser pionero en la conmutación de paquetes a través de redes de radio y canales de satélite. La primera implementación real de *Internet* fue se produjo sobre 1980, cuando DARPA comenzó a convertir las máquinas de su red de trabajo(ARPANET) a los nuevos protocolos de TCP/IP. En 1983 la transición fue completa y DARPA exigió que *todos* los ordenadores que quisieran conectarse a ARPANET usaran TCP/IP.

DARPA contrató además a Bolt, Beranek, y Newman (BNN) para desarrollar una implementación de los protocolos TCP/IP para el UNIX de Berkeley sobre el VAX y dotaron a la Universidad de California en Berkeley para que distribuyese ese código de modo gratuito con su sistema operativo UNIX. El primer lanzamiento de *la distribución del sistema de Berkeley* que incluyó el protocolo TCP/IP estuvo disponible en 1983(BSD 4.2). Desde ese momento, TCP/IP se ha difundido rápidamente entre universidades y centros de investigación y se ha convertido en el estándar de subsistemas de comunicación basados en UNIX. El segundo lanzamiento(BSD 4.3) se distribuyó en 1986, que es actualizado en 1988 (BSD 4.3 Tahoe) y en 1990 (BSD 4.3 Reno). BSD 4.4 fue distribuido en 1993. Debido a limitaciones de fondos, el BSD 4.4 será la última distribución que hará el grupo de investigación de sistemas informáticos("Computer Systems Research Group") de la Universidad de California en Berkeley.

A medida que TCP/IP se extendía rápidamente, nuevas WANs se fueron creando y uniendo a ARPANET en los Estados Unidos. Por otro lado, redes de otros tipos, no necesariamente basadas en TCP/IP, se añadieron al conjunto de redes interconectadas. El resultado fue lo que hoy se conoce como *Internet*. Distintos ejemplos de redes que han jugado papeles clave en este desarrollo se describen en las siguientes secciones.

1.2.1 Internet

¿Qué es exactamente? En primer lugar, la palabra *Internet* es simplemente una contracción de la frase *red interconectada*. Sin embargo, escrita con mayúscula hace referencia a un conjunto mundial de redes interconectadas, de tal forma que Internet es una red interconectada, aunque no a la inversa. A Internet se le llama a veces *"Interred conectada"("connected Internet")*.

Internet está constituida por los siguientes grupos de redes(ver las siguientes secciones para más información):

- Troncales: grandes redes que existen principalmente para interconectar otras redes. Actualmente las redes troncales son NSFNET en US, EBONE en Europa y las grandes redes troncales comerciales.
- Redes regionales que conectan, por ejemplo, universidades y colegios.
- Redes comerciales que suministran acceso a troncales y suscriptores, y redes propiedad de organizaciones comerciales para uso interno que también tienen conexión con Internet.
- Redes locales, como por ejemplo, redes a nivel de campus universitario.

En muchos casos, particularmente en redes de tipo comercial, militar y gubernamental, el tráfico entre estas y el resto de Internet está restringido(ver Cortafuegos("firewalls")). Esto conduce a la pregunta ¿Cómo sé si estoy conectado a Internet? Un enfoque viable es preguntarse: ¿puedo hacer un ping al host *ds.internic.net*? El ping, descrito en Ping, es un programa usado para determinar si un host de una red es alcanzable; está implementado en cualquier plataforma TCP/IP. Si la respuesta es no, entonces no estás conectado. Esta definición no implica necesariamente que uno esté totalmente aislado de Internet: muchos sistemas que fallarían en este test tienen, por ejemplo, pasarelas de correo electrónico a Internet.

1.2.2 ARPANET

Llamado a veces el "abuelito" de las redes de conmutación de paquetes, ARPANET fue construido por DARPA(llamado ARPA en esa época) a finales de los '60 para facilitar la instalación de equipo de investigación de la tecnología de conmutación de paquetes y para permitir compartir recursos a los contratistas del Departamento de Defensa. La red interconectaba centros de investigación, algunas bases militares y emplazamientos gubernamentales. Pronto se popularizó entre los investigadores

mediante la colaboración a través del correo electrónico y de otros servicios. Se desarrolló orientada a una utilidad para la investigación, usada por el DCA("Defense Communications Agency") a finales de 1975 y se dividió en 1983 en MILNET, para la interconexión de localizaciones militares, y ARPANET, para la interconexión de centros de investigación. Esto fue el primer paso hacia la "I" mayúscula de Internet.

En 1974, ARPANET estaba basada en líneas arrendadas de 56kbps que interconectaban *nodos de conmutación de paquetes*(PSN) dispersados por todo US y el oeste de Europa. Eran minicomputadores que ejecutaban un protocolo conocido como*1822* (por el número del informe que lo describía) y dedicados a la tarea de conmutación de paquetes. Cada PSN tenía al menos dos conexiones con otros PSNs(para permitir encaminamiento alternativo en caso de fallo de algún circuito)y hasta 22 puertos para conexiones de ordenadores de usuarios(*hosts*). Los sistemas 1822 permitían la entrega fiable y con control de flujo de un paquete al nodo de destino. Esta es la razón por la que el protocolo *NCP* original fue un protocolo bastante simple. Fue sustituido por los protocolos de TCP/IP, que no asumen la fiabilidad del hardware de red subyacente y pueden ser usados en redes distintas de las basadas en 1822. El 1822 no se convirtió en un estándar de la industria, por lo que posteriormente DARPA decidió reemplazar la tecnología de conmutación de paquetes del 1822 por el estándar *CCITT X.25*.

El tráfico de datos excedió pronto la capacidad de las líneas de 56Kbps que constituían la red, que ya no eran capaces de soportar el flujo requerido. Hoy en día ARPANET ha sido sustituido por las nuevas tecnologías en su papel de troncal en el área de la investigación de Internet(ver NSFNET posteriormente, en este capítulo), mientras que MILNET sigue siendo la red troncal en el área militar.

1.2.3 NSFNET

NSFNET("National Science Foundation Network"), es una red de tres niveles situada en los Estados Unidos y consistente en:

- Una *troncal*: una red que conecta redes de nivel medio administradas y operadas por separado y centros de superordenadores fundados por el NSF. Esta troncal tiene además enlaces transcontinentales con otras redes como por ejemplo EBONE, la red troncal europea de IP.
- *Redes de nivel medio*: de tres clases(regionales, basadas en una disciplina y redes formadas por un consorcio de superordenadores).
- *Redes de campus*: tanto académicas como comerciales, conectadas a las de nivel medio.

La primera troncal.

Establecida originalmente por el NSF("National Science Foundation") como una red de comunicaciones para investigadores y científicos para acceder a los superordenadores del NSF, la primera troncal de NSFNET usaba seis microordenadores DEC LSI/11 como conmutadores de paquetes, interconectados por líneas arrendadas de 56Kbps. Existía una interconexión primaria entre la troncal de NSFNET y ARPANET en el Carnegie Mellon, que permitía el encaminamiento de datagramas entre usuarios conectados a esas redes.

La segunda troncal.

La necesidad de una nueva troncal se manifestó en 1987, cuando la primera quedó sobrecargada en pocos meses(el crecimiento estimado en ese momento fue de un 100% anual). El NSF y MERIT, Inc., un consorcio de redes de ordenadores de ocho universidades estatales de Michigan, acordaron desarrollar y gestionar una nueva troncal de alta velocidad con mayores capacidades de transmisión y de conmutación. Para gestionarla definieron el IS ("*Information Services"*) que está compuesto del Centro de Información y el Grupo de Soporte Técnico. El Centro de Información es responsable de distribuir información, la gestión de recursos informativos y la comunicación electrónica. El grupo de soporte técnico proporciona apoyo técnico directamente sobre el campo de trabajo. El propósito de esto es suministrar un sistema integrado de información con interfaces fáciles de usar y administrar, accesible desde cualquier punto de la red y apoyado por toda una serie de servicios de formación.

MERIT y NSF dirigieron este proyecto con IBM y MCI. IBM proporcionó el software, equipo para la conmutación de paquetes y la gestión de redes, mientras que MCI aportó la infraestructura para el transporte a largas distancias.

Instalada en 1988, la nueva red usaba inicialmente circuitos arrendados de 448Kbps para interconectar 13 *sistemas nodales de conmutación*(NSS) suministrados por IBM. Cada NSS estaba compuesto de nueve sistemas RT de IBM(que usaban una versión IBM del BSD 4.3) conectados a través de dos redes en anillo de IBM(por redundancia). En cada una de las 13 localizaciones se instaló un IDNX("Integrated Digital Network Exchange") de IBM, para permitir:

- Encaminamiento dinámico alternativo
- Reserva dinámica de ancho de banda

La tercera troncal

En 1989, la topología de los circuitos de NFSNET fue reconfigurada después de haber medido el tráfico y la velocidad de las líneas arrendadas se incrementó a T1(1.544Mbps) usando principalmente fibra óptica.

Debido a la necesidad constantemente creciente de mejoras en la conmutación de paquetes y en la transmisión, se añadieron tres NSSs a la troncal y se actualizó la velocidad de las conexiones. La migración de NFSNET de T1 a T3(45Mbps) se completo a finales de 1992. Advanced Network & Services, Inc. (compañía fundada por IBM, MCI, Merit, Inc.) es en la actualidad el organismo proveedor y gestor de NFSNET.

La migración a niveles de gigabits ya ha empezado y continuará durante finales de l990. Para más información, remitirse a Futuro.

El gobierno de US pretende retirar sus fondos de NSFNET en abril de 1995. Esto es parte de una reacción ante el uso comercial de NSFNET. Para más detalles sobre este tema, ver Uso comercial de Internet y La super autopista de la información respectivamente.

1.2.4 EBONE

EBONE,("Pan-European Multi-Protocol Backbone") juega en el tráfico de Internet en Europa el mismo papel que NSFNET en US. EBONE tiene conexiones a nivel de kilobit y megabit entre cinco grandes centros.

1.2.5 CREN

Completado en octubre de 1989, el organismo fusionador de las dos famosas redes CSNET("Computer+Science Network") y BITNET(*"Because It's Time* Network") formó el CREN("Corporation for Research and Educational Networking"). CREN abarca la familia de servicios históricas de CSNET y BITNET para proporcionar una rica variedad de opciones en la conexión de redes:

PhoneNet
: Es el servicio original de red de CSNET y proporciona servicio de correo electrónico "store-and-forward" usando líneas telefónicas de marcaje(1200/2400 bps). Permite a los usuarios intercambiar mensajes con otros miembros del CREN y de otras grandes redes de correo, incluyendo a NSFNET, MILNET, etc.

X.25Net
: Es un red de CSNET conectada a Iternet que suministra un servicio completo, usa protocolo TCP/IP sobre X.25. Es habitual que los miembros internacionales se conecten a CSNET, ya que pueden usar su red pública de datos("Public Data Network") X.25 para alcanzar a Telenet en US. Aporta transferencia de ficheros, telnet, así como servicio inmediato de correo electrónico entre host de X.25Net.

IP de marcaje
: Es una implementación de SLIP("Serial Line IP") que permite que los sitios que usan la red telefónica conmutada(9600bps) envíen paquetes IP, por medio de un servidor central, a Internet. Los usuarios de este método tienen los mismos servicios que en X.25Net.

Línea IP arrendada
: Usada por muchos miembros del CREN para conectarse a CREN. Soporta una serie de velocidades de enlace hasta tasas T1.

RSCS/NJE sobre BISYNC
: Tradicionalmente funciona sobre líneas arrendadas a 9600bps y proporciona servicio de mensajes interactivos, transferencias de ficheros no solicitadas y correo electrónico.

RSCS sobre IP
: Permite a los "hub" del servicio BITNET relajar las líneas dedicadas de RSCS BYSYNC en favor de una ruta IP, si existe.

1.2.6 Cypress

Cypress es una red sobre líneas arrendadas que permite tener un sistema de conmutación de paquetes de bajo coste e independiente del protocolo, usado principalmente para interconectar sitios pequeños a redes de Internet sobre TCP/IP. Establecido en origen como parte de un proyecto de investigación conjunto con CSNET, ahora es independiente de CSNET.

No hay restricciones sobre su uso, aparte de las impuestas por otras redes. De este modo el tráfico comercial puede pasar entre dos sitios industriales a través de Cypress. Los sitios industriales no pueden pasar tráfico comercial sobre Internet debido a restricciones impuestas por agencias gubernamentales que controlan las redes troncales (por ejemplo, NFSET).

1.2.7 DRI

TWN("Terrestrial Wideband Network") o Red Terrestre de Banda Ancha es una WAN con el propósito de proporcionar una plataforma para la investigación con protocolos y aplicaciones en redes de alta velocidad(papel representado inicialmente por ARPANET). Este sistema incluye servicios tanto orientados a conexión como no orientados a conexión, broadcast y conferencia en tiempo real.

La TWN fue construida y puesta en marcha por BNN Systems y Technologies Corporation durante la primera mitad de 1989 como parte de la fase inicial del DRI("Defense Research Internet"). Su principal finalidad era transportar a lo largo y ancho del

país el tráfico de datagramas asociado a proyectos subvencionados por DARPA. Estaba compuesto de pasarelas de Internet y conmutadores de paquetes WPSs("Terrestrial Wideband Network packet switches") que se comunicaban entre sí usando el HAP("Host Access Protocol") especificado en RFC 1221. Se usó el WB-MON("Wideband Monitoring Protocol") entre los WPSs y el centro de monitorización. La troncal soportaba también un entorno de investigación para conferencia multimedia y conferencia con voz y vídeo empleando pasarelas que utilizaban un protocolo orientado a conexión en tiempo real(ST-II - Stream Protocol - RFC 1190) sobre un red no orientada a conexión.

1.2.8 EARN("European Academic Research Network")

EARN, iniciada en 1983, fue la primera y mayor red en dar servicio a instituciones académicas y de investigación en Europa, Oriente Medio y África. EARN comenzó su andadura con la ayuda de IBM. Evolucionó para convertirse en una red sin ánimo de lucro y basada en tráfico no comercial que da servicio a instituciones académicas y de investigación.

1.2.9 RARE("Réseaux Associés pour la Recherche")

RARE, fundada en 1986, es la asociación de organizaciones de redes europeas y sus usuarios. La asociación tiene 20 *FNM("Full National Members";* todos países europeos), numerosos *ASN("Associate National Members";* algunos países europeos y asiáticos), *IM("International Members";* por ejemplo EARN) y *LM("Liason Members";* por ejemplo CREN).

Soporta los principios de los sistemas abiertos tal como se definen en ISO además de un número de grupos principalmente europeos, como el EWOS *("European Workshop for Open Systems")* y el ETSI *("European Telecommunications Standards Institute")* .

Para más detalles, remitirse a RARE*("Réseaux Associés pour la Recherche Européenne")*.

1.2.10 RIPE("Réseaux IP Européens")

El "Réseaux IP Européens"(RIPE) coordina las redes TCP/IP para la comunidad científica en Europa. Opera bajo los auspicios de RARE. RIPE lleva funcionando desde 1989. A comienzos de los '90 más de 60 organizaciones participaban en este trabajo. El objetivo de RIPE es asegurar la coordinación administrativa y técnica necesaria para permitir el funcionamiento de la red *IP* pan-Europea. Notar que RIPE *no* gestiona ninguna red de su propiedad. RIPE puede definirse como la actividad *IP* de RARE.

Una de las actividades de RIPE es, mantener una base de datos de redes IP europeas, dominios DNS y sus contactos. El contenido de esta base de datos se considera de dominio público. La base de datos puede ser accedida vía un servidor *WHOIS* en el host *whois.ripe.net* (puerto TCP 43) o vía un FTP anónimo a *ftp.ripe.net*.

El RIPE NCC ("Network Coordination Center") se puede conectar vía:
RIPE NCC
Kruislaan 409
NL-1098 SJ Amsterdam
The Netherlands
Phone: +31 20 592 5065
Fax: +31 20 592 5155
E-mail: ncc@ripe.net

1.2.11 Internet en Japón

Japón tiene muchas redes distintas. Las siguientes son algunas de las principales.

- La BITNET japonesa comenzó a funcionar en 1985. Fue fundada por la Universidad de la Ciencia de Tokyo y parte de sus miembros. Esta red conecta con CUNY ("City University of New York") a través de un enlace a 56 Kbps.
- N-1net es gestionada por el NACSIS *("National Center for Science and Information Systems")*, un instituto de investigación fundado por el Ministerio de Educación de Japón. Empezó a funcionar en 1980 usando una red de conmutación de paquetes X.25. N-1net tiene una conexión de 50 Kbps con el NSF en Washington.
- EL TISN("International Science Network") de Todai es usado por físicos y químicos. TISN tiene un enlace de 128 Kbps entre Todai y Hawaii.
- WIDE*("Widely Integrated Distributed Environment")* es la versión japonesa de Internet. Comenzó como un proyecto de investigación en 1986. Hay dos conexiones entre WIDE y el resto de Internet. Uno, de 192 Kbps va de la Universidad de Keio en Fujisawa a la Universidad de Hawaii. El otro es un enlace secundario de 128 Kbps de Todai a Hawaii, previsto

para el caso de que falle el principal.

Para más detalles, remitirse a [LaQuey] y [Malamud] listados en Bibliografía.

1.2.12 Uso comercial de Internet

En años recientes Internet ha crecido en tamaño y extensión a una ritmo mayor de lo que nadie podría haber previsto. En particular, más de la mitad de los hosts conectados hoy a Internet son de carácter comercial. Esta es un área conflictiva, potencial y realmente, con los objetivos iniciales de Internet, que eran favorecer y cuidar del desarrollo de las comunicaciones abiertas entre instituciones académicas y de investigación. Sin embargo, el crecimiento continuado en el uso comercial de Internet es inevitable por lo que será útil explicar como está teniendo lugar esta evolución.

Una iniciativa importante a tener en cuenta es la de AUP*("Acceptable Use Policy")*. La primera de estas políticas se introdujo en 1992 y se aplica al uso de NSFNET. Una copia de ella se puede conseguir en *nic.merit.edu/nsfnet/acceptable.use.policies*. En el fondo AUP es un compromiso "para apoyar la investigación y la educación abierta". Bajo "usos inaceptables" está la prohibición de "uso para actividades lucrativas", a menos que se hallen incluidas en el Principio General o como un uso específico aceptable. Sin embargo, a pesar de estas instancias aparentemente restrictivas, NSFNET se ha ido usando cada vez más para un amplio abanico de actividades, incluyendo muchas de naturaleza comercial.

Aparte del AUP de NSFNET, muchas de las redes conectadas a NSFNET mantienen sus propios AUPs. Algunos de ellos son relativamente restrictivos en su tratamiento de las actividades comerciales mientras que otros son relativamente liberales. Lo importante es que los AUP tendrán que evolucionar mientras continúe el inevitable crecimiento comercial en Internet.

Concentrémonos ahora en los proveedores de servicios en Internet que han desarrollado mayor actividad en la introducción de usos comerciales de Internet. Dos dignos de mencionar son PSINet y UUNET, que a finales de los '80 comenzaron a ofrecer acceso a Internet tanto a negocios como a individuos. CERFnet, establecida en California, ofrece servicios libres de cualquier AUP. Poco después se formó una organización para unir PSINet, UUNet y CERFNet, llamada CIX("Commercial Internet Exchange"). Hasta la fecha CIX tiene más de 20 miembros que conectan las redes constituyentes en un entorno libre de AUPs. Sobre el mismo momento en que surgió CIX, una compañía sin ánimo de lucro, ANS("Advance Network and Services "), fue formada por IBM, MCI y Merit,Inc. con el fin de operar conexiones troncales T1 para NSFNT. Este grupo ha permanecido activo e incrementando su presencia comercial en Internet.

ANS formó también una subsidiaria orientada comercialmente denominada ANS CO+RE para proporcionar enlaces entre clientes comerciales y dominios educacionales y de investigación. ANS CO+RE suministra además acceso libre de AUPs a NSFNET al estar conectada a CIX.

1.2.13 La super autopista de la información

Una reciente e importante iniciativa ha sido la creación del Consejo Estadounidense de Asesoramiento sobre la Infraestructura Nacional de Información("US Advisory Council on the National Information Infrastructure") dirigido por Al Gore. En esencia, la iniciativa hace de la creación de una "red de redes" una prioridad nacional. Esta red debería ser similar a Internet en algunos aspectos, pero con el gobierno y la industria contribuyendo cada uno con lo mejor de sí mismo.

Desde una perspectiva más internacional, el Grupo de los Siete("The Group of Seven(G7)") ministros se reunió en Bruselas en febrero de 1995 para discutir sobre la incipiente GII("Global Information Infrastructure"). Los ministros de tecnología y economía de Canadá, el Reino Unido, Francia, Japón. Alemania, Italia, y los Estado Unidos acudieron a la conferencia, y se concentraron en las implicaciones tecnológicas, culturales y económicas concernientes al desarrollo de la infraestructura nacional.

Una revista electrónica gratuita llamada G7 Live se utilizó para hacer llegar diariamente a los usuarios de Internet los comentarios y noticias sobre la conferencia. Aspectos específicos cubiertos por G7 Live incluyen los derechos de la propiedad intelectual, construcción de infraestructuras, consideraciones culturales y legislativas, y descripciones de las más de 100 exhibiciones tecnológicas presentes en la conferencia.

Tanto el NII como el GII descritos anteriormente son iniciativas importantes que en última instancia deberían conducir a la "super autopista de la información" que es en la actualidad el objeto de tanta discusión en los medios de comunicación.

Tabla de contenidos Futuro

1.4 Futuro

La perspectiva a largo plazo descrita en el HPCC (*"United States Federal High Performance Computing and Communications Program")* indica que todas las redes de Internet serán absorbida por el NREN (*"National Research and Education Network"*).

El HPCA(*"High Performance Computing Act"*) de 1991 fue legalizada en Estados Unidos en diciembre de 1991. Establecía un presupuestos de unos 100 millones de dólares anuales durante 5 años.

El desarrollo de la red del programa NREN se ha orientado a sistemas de computación distribuidos para instituciones educativas y de investigación, así como a investigaciones en redes y aplicaciones de alta velocidad.

NREN ya ha organizado la integración, en coordinación con el NSF, del DRI ("DoD's Defense Research Internet"), NSFNet, la NASA ("National Aeronautics and Space Administration"), el NSI("Science Internet"), el DOE("Department of Energy") y Esnet("Energy Science Network").

El programa NREN especifica un proyecto de tres fases dirigido por DARPA para incrementar la velocidad de transmisión de datos a 3 Gbps durante los próximos 10 – 15 años. Este programa también incluye la exploración de mecanismos de tasación de servicios de red y el comienzo de una *transición estructurada a los servicios comerciales*.

En agosto de 1992, bajo el programa NREN, el DOE firmó un contrato de 5 años por valor de 50 millones de dólares con la Sprint Corporation para servicios públicos de ATM .

En el momento de redactar este documento, la evolución de NREN en lo que respecta alas redes de alta velocidad es considerable.

El NAP("*Network Access Point")* es un cambio importante de cara a la nueva arquitectura de Internet. El NSF ha escogido una serie de organizaciones para que operen el NAP. Entre ellas están:

- Sprint Corporation - Nueva York/New Jersey
- Ameritech - Chicago
- PacBell - San Francisco/Bay Area
- MFSdatanet - Washington DC

El NSP("*Network Service Provider")* es un proveedor de servicios de Internet capacitado para disponer de conectividad local gracias al NSF. Esto significa que el NSP se debe conectar a tres NAPs primarios en California, Chicago y Nueva York. Entre estos se hallan:

- ANS (ahora propiedad de America Online)
- MCInet
- SprintLink

Además, hay un número de ISPs("*Internet Service Providers")* que no tienen esta conectividad local. Entre ellas están:

- AlterNet
- Net99
- PSI

Un RA("*Routing Arbitor")* es una organización que recibe fondos del NSF y que proporciona información de encaminamiento a cada NAP.

Los NAPs proporcionan actualmente servicios de alta velocidad basados en ATM, retransmisión de trama y FDDI a los NSPs y los ISPs.

Para más información sobre estos NAPs, remitirse a *http://rrdb.merit.edu/napsp3.html http://rrdb.merit.edu/pacbell.html*.

1.4.1 Futuro – Redes de alta velocidad

El futuro de NREN está influenciado, cuando menos, por los avances en la tecnología de redes de alta velocidad.

- La **retransmisión de tramas** es un estándar de red que sirve de interfaz en redes orientadas a paquetes. Soporta tamaños de paquete variables, por lo que no está recomendado para el tráfico isócrono, como por ejemplo de voz o vídeo. Una red se puede implementar fácilmente con un equipo ya existente de conmutación de paquetes empleando la retransmisión de tramas para mejorar su rendimiento. La velocidad actual de esta tecnología es T - 1 (1.544 Mbps).
- **DQDB (Distributed Queue Dual Bus)** es un protocolo diseñado para manejar tráfico tantro isócrono como de datos sobre enlaces de ópticos de alta velocidad. Los medios de transmisión definidos son:
  - *Conexión de fibra* a 35 Mbps y 155 Mbps
  - *SONET (Synchronous Optical Network)* a partir de 51,840 Mbps.

  Fue aceptado como estándar (IEEE 802.6) por las MANs ("Metropolitan Area Networks").

- **ATM (Asynchronous Transfer Mode)**, es una tecnología de conmutación basada en celdas de longitud fija de 53 bytes. Proporciona velocidades altas (definidas a 155 Mbps y 622 Mbps) y es adecuada para la transmisión de voz, vídeo y datos. No se espera que las redes públicas soporten ATM hasta finales de los ´90.
- **ISDN de banda ancha** es un tecnología nueva, aunque no estandarizada, que ofrece velocidades incluso mayores, a partir de OC - 3 ("Optical Carrier level 3", 155.52 Mbps). No se espera que esté disponible hasta 1995, como pronto.

Tabla de contenidos RFC(Request For Comments)

1.5 RFC("Request For Comments")

La pila de protocolos de Internet sigue evolucionando mediante el mecanismo conocido como *RFC("Request For Comments")*. Los investigadores están diseñando e implementando nuevos protocolos(en su mayoría del nivel de aplicación), que se ponen en conocimiento de la comunidad de Internet en la forma de un RFC. (0)El RFC es descrito por el IAB("Internet Architecture Board"). La mayor fuente de RFCs es el IETF("Internet Engineering Task Force") que es una organización subsidiaria del IAB. Sin embargo, cualquiera puede enviar un informe propuesto como RFC al editor de los RFC. Hay una serie de normas que los autores de RFCs deben seguir para que su RFC sea aceptado. Estas reglas se describen en un RFC(RFC 1543) que además indica como enviar una propuesta de RFC.

Una vez que un RFC ha sido publicado, todas las revisiones y sustituciones se publican como nuevos RFCs. Se dice que un nuevo RFC que revisa o sustituye un RFC ya existente "actualiza" o "desfasa" a ese RFC. Asimismo, el RFC original es "actualizado" o "desfasado" por el nuevo. Por ejemplo, el RFC 1521 que describe el protocolo MIME es una "segunda edición", siendo una revisión del RFC 1341, y el RFC 1590 es una enmienda del 1521. Por tanto el RFC 1521 se etiqueta del modo siguiente: "Deja obsoleto al RFC 1341; Actualizado por el RFC 1590". En consecuencia, nunca hay confusión sobre si dos personas se refieren a dos versiones distintas de un RFC,

Algunos RFCs se califican como *documentos informativos* mientras que otros describen protocolos de Internet. El IAB("Internet Architecture Board") mantiene una lista de todos los RFCs que describen la pila de protocolos. A cada uno de ellos se le asigna un estado y un status.

Todo protocolo Internet puede tener uno de los siguientes estados:

Estándar
: El IAB lo ha establecido como protocolo oficial de Internet. Se dividen en dos grupos:
  2. El protocolo y superiores, protocolos que se aplican a la totalidad de Internet.
  4. Protocolos específicos de redes, generalmente especificaciones del funcionamiento de IP en tipos concretos de redes.

Estándar provisional
: El IAB está considerando activamente este protocolo como un posible protocolo estándar. Es deseable disponer de comentarios y pruebas exhaustivas cuantitativa y cualitativamente. Los comentarios y los resultados de las pruebas deberían enviarse al IAB. Existe la posibilidad de que se efectúen cambios en un protocolo estándar antes de que se convierta en estándar.

Propuesto como estándar.
: Se trata propuestas de protocolos que el IAB puede considerar para la estandarización en el futuro. Es deseable evaluar la implementación y el testeo sobre un gran número grupos. Es probable que el protocolo se someta a revisión.

Experimental
: Un sistema no debería implementar un protocolo experimental a menos que participe en el experimento y haya coordinado el uso que va a hacer del protocolo con el que lo ha desarrollado.

Informativo
: Los protocolos desarrollados por otras organizaciones de estándares, o distribuidores, o aquellos que por otras razones son ajenos a los propósitos del IAB, pueden ser publicados a conveniencia de la comunidad de Internet como protocolos informativos. En algunos casos el IAB puede recomendar el uso de estos protocolos en Internet.

Histórico
: Son protocolos con pocas posibilidades de convertirse alguna vez en estándar en Internet, bien porque han quedado desfasados por protocolos posteriores o debido a la falta de interés.

Definiciones de los status de los protocolos:

Requerido
: Un sistema debe implementar los protocolos requeridos.

Recomendado
: Un sistema debería implementar un protocolo recomendado.

Electivo
: Un sistema puede o no implementar un protocolo electivo. La idea general es que si vas a implementarlo, debes hacerlo exactamente como se define.

Uso limitado.
: Estos protocolos son usados en circunstancias específicas. Esto se puede deber a su estado experimental, naturaleza específica, funcionalidad limitada o estado histórico.

No recomendado.
: Estos protocolos no se recomiendan para el uso general. Esto se puede deber a su limitada funcionalidad, naturaleza específica, o a que su estado es experimental o histórico.

1.5.1 Estándares de Internet

Los estándares propuestos, provisionales, y los protocolos estándar figuran en el "Internet Standards Track"("Seguimiento de estándares de Internet"). El seguimiento de estándares es controlado por el *IESG("Internet Engineering Steering Group")* del IETF. Cuando un protocolo alcanza el estado de estándar, se le asigna un número de estándar(STD). El propósito del STD es indicar claramente que RFCs describen estándares de Internet. Los números STD referencian múltiples RFCs cuando la especificación de un estándar está repartida entre varios documentos. A diferencia de los RFCs, donde el número se refiere a un documento específico, los números STD no cambian cuando un estándar es actualizado. Sin embargo, los STD carecen de número de versión ya que todas las actualizaciones se hacen a través de RFCs y los RFCs son únicos. De este modo, para especificar sin ambigüedades a que estándar se refiere uno, el número de estándar y todos los RFCs que incluye deberían ser mencionados. Por ejemplo, el DNS("Domain Name System") tiene el STD 13, y se describe en los RFCs 134 y 1035. Para referenciar un estándar, se debería usar una forma como "STD-13/RFC-1034/RFC-1035". Para una descripción de los procedimientos para estándares, remitirse al *RFC 1602 -- Los procedimientos para estándares de Internet - Revisión 2*.

Para el seguimiento de algunos estándares, el status del RFC no siempre contiene suficiente información como para ser útil. Por ello se le añade un *descriptor de aplicabilidad*, dado bien en la forma de STD 1 en un RFC separado; este descriptor lo dan particularmente los protocolos de encaminamiento.

En este documento se hacen referencias a RFCs y número STD, ya que constituyen la base de todas las implementaciones de protocolos TCP/IP.

Cuatro estándares de Internet son de particular importancia:

STD 1 - Estándares de protocolo oficiales en Internet
: Este estándar da el estado y el status de cada estándar o protocolo de Internet, y define los significados atribuidos a cada estado o status. El IAB suele emitirlo aproximadamente cada trimestre. En el momento de escribir este documento, este estándar va por el RFC 1780 (marzo de 1995).

STD 2 - Números asignados de Internet
: Este estándar lista los número asignados actualmente y otros parámetros de protocolos en la pila de protocolos de Internet. Es emitido por IANA("Internet Assigned Numbers Authority"). La edición actual en el momento de escribir este documento se corresponde con el RFC 1700 (octubre de 1994).

STD 3 - Requerimientos de host
: Este estándar define los requerimientos para el software de Internet del host (con frecuencia a través de referencias a RFCs importantes). El estándar aparece dividido en dos partes: el *RFC 1122 - Requerimientos para hosts en Internet - de la capa de comunicaciones* y el *RFC 1123 - Requerimientos para hosts en Internet - de aplicación y soporte*.

STD 4 - Requerimientos de pasarela
: Este estándar define los requerimientos para el software de pasarelas. Su RFC es el 1009.

1.5.2 Para Su Información("For Your Information(FYI)")

Cierto número de RFCs que tienen un amplio interés para los usuarios de Internet se clasifican como documentos *FYI("For Your Information")*. Frecuentemente contienen información de ayuda o de carácter introductorio. Como los números STD, un FYI no se cambia cuando se publica un RFC revisado. A diferencia de los STDs, los FYIs corresponden a un único RFC. Por ejemplo, el *FYI 4 -- FYI acerca de preguntas y respuestas - Respuesta a preguntas habituales de nuevos usuarios de Internet* va en la actualidad por su cuarta edición. Los números de RFC son 1177, 1206, 1325 y 1594.

1.5.3 Obteniendo RFCs

Todos los RFCs están disponibles para el público, en forma de documento tanto impreso como electrónico, por medio del Internic("Internet Network Information Center"; internic.net). Antes de 1993, el DNN NIC(*nic.ddn.mil*) realizaba la función del NIC. Consultar el RFC 1400 para tener más información acerca de esta transición.

- Los RFCs pueden conseguirse en forma impresa de:
  Network Solutions, Inc.
  Attn: InterNIC Registration Service
  505 Huntmar Park Drive
  Herndon, VA 22070

  Help Desk Telephone Number:
  703-742-4777

  FAX Number 703-742-4811
- Para conseguir el documento electrónico, los usuarios pueden hacer un FTP anónimo a *ds.internic.net (198.49.45.10)* y tomar los ficheros del directorio rfc, o un Gopher a *internic.net (198.41.0.5)*.
- Para información sobre otros métodos de acceder a RFCs vía E-mail o FTP, envía un E-mail a *"rfc-info@ISI.EDU"* con el mensaje *"help: ways_to_get_rfcs"*. Por ejemplo:

```
To: rfc-info@ISI.EDU
Subject: getting rfcs

help: ways_to_get_rfcs
```

- Si tienes acceso a Internet, hay muchos sitios que mantienen archivos de RFCs. Uno que podrías probar es el *"MAGIC Document Archive"* en *http://www.msci.magic.net/docs/rfc/rfc_by_num.html*.
- Los RFCs también se pueden obtener a través de la red IBM VNET usando el siguiente comando:

```
EXEC TOOLS SENDTO ALMVMA ARCNET RFC GET RFCnnnn TXT *
```

Donde nnnn es el número del RFC.

Para conseguir una lista de todos los RFCs(y saber si están disponibles en formato TXT o postscript), usa el comando:

```
EXEC TOOLS SENDTO ALMVMA ARCNET RFC GET RFCINDEX TXT *
```

También están los archivos STDINDEX TXT y FYIINDEX TXT que listan aquellos RFCs que tienen un número ST o FYI.

1.5.4 Principales protocolos de Internet

Para dar una idea de la importancia de los principales protocolos, listamos algunos de ellos junto con su estado actual, status y STD donde es aplicable en Tabla - Estado, status y números STD actuales de protocolos importantes de Internet. La lista completa se puede encontrar en RFC 1780 - Estándares de protocolos oficiales en Internet. **Leyenda:**
Estado: Std. = Estándar; Draft = Estándar provisional; Prop. = Propuesto como estándar; Info. = Informativo; Hist. = Histórico
Status: Req. = Requerido; Rec. = Recomendado; Ele. = Electivo; Not = No Recomendado

Protocol	Name	State	Status	STD
IP	Internet Protocol	Std.	Req.	5
ICMP	Internet Control Message Protocol	Std.	Req.	5
UDP	User Datagram Protocol	Std.	Rec.	6
TCP	Transmission Control Protocol	Std.	Rec.	7
TELNET	TELNET Protocol	Std.	Rec.	8
FTP	File Transfer Protocol	Std.	Rec.	9
SMTP	Simple Mail Transfer Protocol	Std.	Rec.	10
MAIL	Format of Electronic Mail Messages	Std.	Rec.	11
DOMAIN	Domain Name System	Std.	Rec.	13
DNS-MX	Mail Routing and the Domain System	Std.	Rec.	14
MIME	Multipurpose Internet Mail Extensions	Draft	Ele.	
SNMP	Simple Network Management Protocol	Std.	Rec.	15
SMI	Structure of Management Information	Std.	Rec.	16
MIB-I	Management Information Base	Hist.	Not	
MIB-II	Management Information Base-II	Std.	Rec.	17
NETBIOS	NetBIOS Services Protocol	Std.	Ele.	19
TFTP	Trivial File Transfer Protocol	Std.	Ele.	33
RIP	Routing Information Protocol	Std.	Ele.	34
ARP	Address Resolution Protocol	Std.	Ele.	37

TFTP	Trivial File Transfer Protocol	Std.	Ele.	33
RIP	Routing Information Protocol	Std.	Ele.	34
ARP	Address Resolution Protocol	Std.	Ele.	37
RARP	Reverse Address Resolution Protocol	Std.	Ele.	38
GGP	Gateway to Gateway Protocol	Hist.	Not	
BGP3	Border Gateway Protocol 3	Draft	Ele.	
OSPF2	Open Shortest Path First Protocol V2	Draft	Ele.	
IS-IS	OSI IS-IS for TCP/IP Dual Environments	Prop.	Ele.	
BOOTP	Bootstrap Protocol	Draft	Rec.	
GOPHER	The Internet Gopher Protocol	Info.		
SUN-NFS	Network File System Protocol	Info.		
SUN-RFC	Remote Procedure Call Protocol Version 2	Info.		

Tabla: Estado, status y números STD actuales de protocolos importantes de Internet

En el momento de escribir este documento, no hay ningún RFC asociado al protocolo de transferencia de hipertexto("HTTP") usado en implementaciones de la "World Wide Web". Sin embargo, el documento HyperText Transfer Protocol (HTTP) escrito por Tim Berners-Lee se puede obtener en *ftp://info.cern.ch/pub/www/doc/http-spec.text*.

Adicionalmente, los siguientes RFCs describen el URL("Uniform Resource Locator") y conceptos asociados a él:

- RFC 1630 - Identificadores universales de recursos en WWW
- RFC 1737 - Requerimientos funcionales para los URN("Uniform Resource Names")
- RFC 1738 - URL("Uniform Resource Locators")

Tabla de contenidos Architectura y protocolos

Capítulo 2. Arquitectura y protocolos

En este capítulo comenzaremos con una introducción a TCP/IP y describiendo sus propiedades básicas, tales como la formación de redes, la distribución de protocolos por capas y el encaminamiento. A continuación discutiremos cada uno de los protocolos específicos en detalle.

2.1 Modelo arquitectónico

La pila TCP/IP se llama así por dos de sus protocolos más importantes: : TCP("Transmission Control Protocol") de IP("Internet Protocol"). Otro nombre es pila de protocolos de Internet, y es la frase oficial usada en documentos oficiales de estándares. En este manual utilizaremos el término TCP/IP, que es más habitual.

2.1.1 Redes

La primera meta de diseño de TCP/IP fue construir una interconexión de redes que proporcionase servicios de comunicación universales: una *red*, o *internet*. Cada red física tiene su propia interfaz de comunicaciones dependiente de la tecnología que la implementa, en la forma de una interfaz de programación que proporciona funciones básicas de comunicación(primitivas). Las comunicaciones entre servicios las proporciona el software que se ejecuta entre la red física y la aplicación de usuario, y da a estas aplicaciones una interfaz común, independiente de la estructura de la red física subyacente. La arquitectura de las redes físicas es transparente al usuario.

El segundo objetivo es *interconectar* distintas redes físicas para formar lo que al usuario le parece una única y gran red. Tal conjunto de redes interconectadas se denomina *"internetwork"* o *internet*.

Para poder interconectar dos redes, necesitamos un ordenador que esté conectado a ambas redes y que pueda retransmitir paquetes de una a la otra; tal máquina es un *"router"*. El término *"router" IP* también se usa porque la función de encaminamiento es parte de la capa IP de la pila TCP/IP(Ver protocolos por capas).

Figura - Ejemplos muestra dos ejemplo de "internetworks".

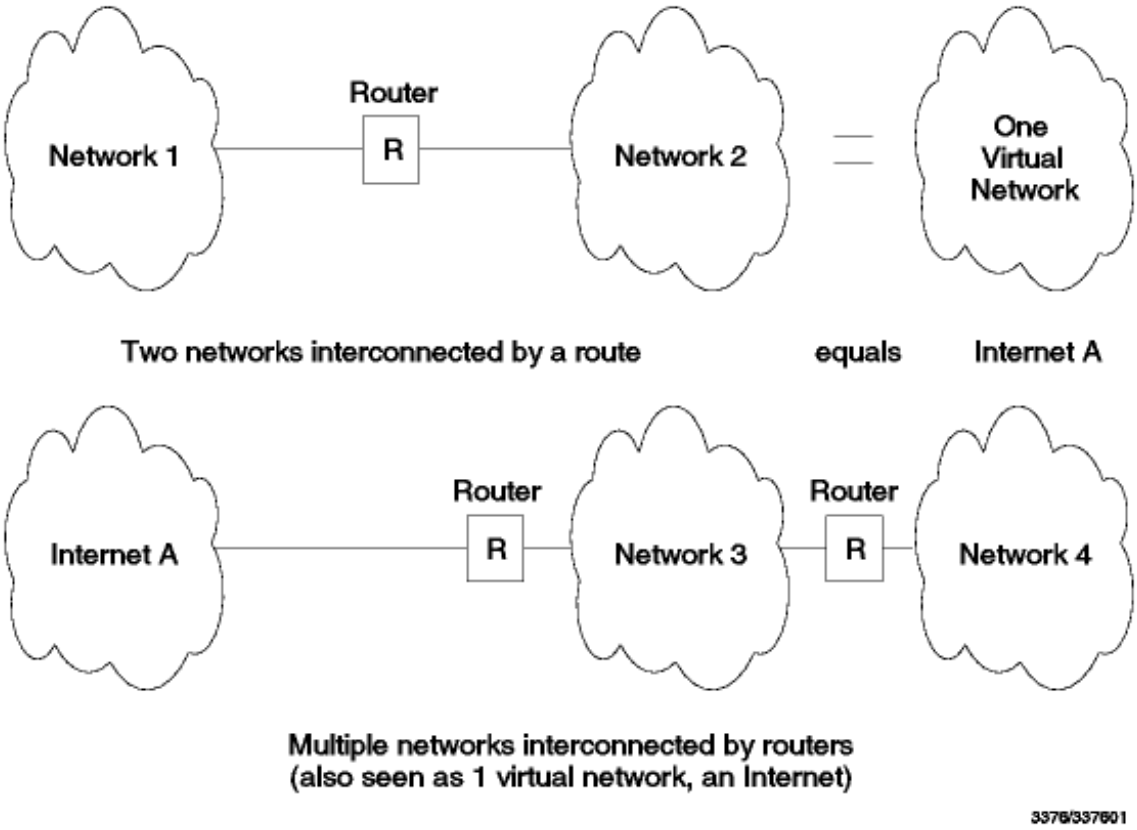


Figura: Ejemplos - Dos conjuntos interconectados de redes, cada uno visto como una red lógica.

Las propiedades básicas de un "router" son:

- Desde el punto de vista de la red, es un host normal.
- Desde el punto de vista del usuario, es invisible. El usuario sólo ve una gran red.

Para ser capaz de identificar un host en la red, a cada se le asigna una dirección, la *dirección IP*. Cuando un host tiene múltiples adaptadores de red, cada adaptador tiene una dirección IP separada. La dirección IP consta de dos partes:

```
dirección IP  = <número de red><número de host>
```

El *número de red* lo asigna una autoridad central y es unívoco en Internet. La autoridad para asignar el *número de host* reside en la organización que controla la red identificada por el número de red. El esquema de direccionamiento se describe en detalle en Direccionamiento.

2.1.2 Arquitectura de Internet

La pila TCP/IP ha evolucionado durante unos 25 años. Describiremos algunos de sus aspectos mas importantes en los siguientes capítulos.

2.1.2.1 Protocolos por capas

TCP/IP, como la mayoría del software de red, está modelado en capas. Esta representación conduce al término *pila de protocolos*. Se puede usar para situar(pero *no* para comparar funcionalmente) TCP/IP con otras pilas, como SNA y OSI("Open System Interconnection"). Las comparaciones funcionales no se pueden extraer con facilidad de estas estructuras, ya que hay diferencias básicas en los modelos de capas de cada una.

Los protocolos de Internet se modelan en cuatro capas:

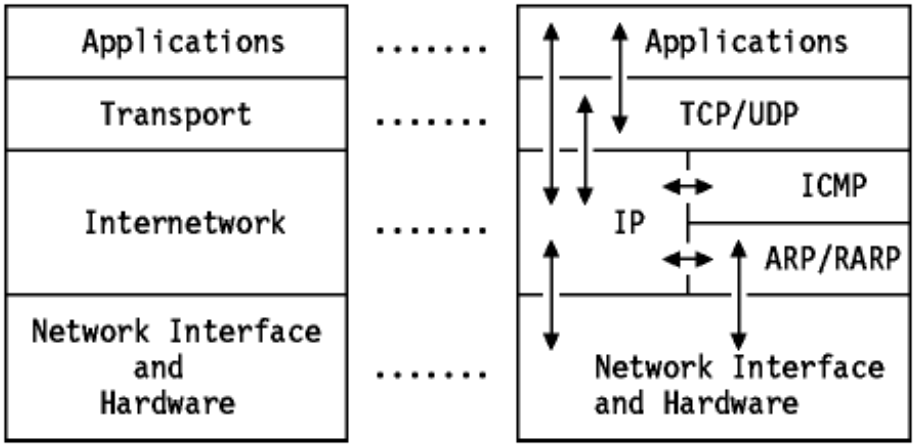


Figura: Modelo arquitectónico - Cada capa representa un ":q.package:eq." de funciones.

Aplicación
: es a un proceso de usuario que coopera con otro proceso en el mismo o en otro host. Ejemplos son TELNET (un protocolo para la conexión remota de terminales), FTP ("File Transfer Protocol") y SMTP ("Simple Mail Transfer Protocol"). Estos se discuten con más detalle en protocolos de aplicación.

Transporte
: proporciona la transferencia de datos de entre los extremos. Ejemplo son TCP(*orientado a conexión*) y UDP(*no orientado a conexión*). Ambos se discuten en detalle en TCP("Transmission Control Protocol") y UDP("User Datagram Protocol")

"Internetwork"
: también llamada *capa de red*, proporciona la imagen de "red virtual" de Internet(es decir, oculta a los niveles superiores la arquitectura de la red). IP("Internet Protocol") es el protocolo más importante de esta capa. Es una protocolo *no orientado a conexión que no asume la fiabilidad de las capas inferiores.* No suministra fiabilidad, control de flujo o recuperación de errores. Estas funciones debe proporcionarlas una capa de mayor nivel, bien de transporte con TCP, o de aplicación, si se utiliza UDP como transporte. IP se discute con detalle en IP("Internet Protocol"). Una unidad de un mensaje en una red IP se denomina datagrama IP. Es la unidad básica de información transmitida en redes TCP/IP networks. Se describe en El datagrama IP.

Network Interface
: o *capa de enlace* o *capa de enlace de datos*, constituye la interfaz con el hardware de red. Esta interfaz puede proporcionar o no entrega fiable, y puede estar orientada a flujo o a paquetes. De hecho, TCP/IP no especifica ningún protocolo aquí, pero puede usar casi cualquier interfaz de red disponible, lo que ilustra la flexibilidad de la capa IP. Ejemplos son IEEE 802.2, X.25 (que es fiable por sí mismo), ATM, FDDI, PRN("Packet Radio Networks", como AlohaNet) de incluso SNA.

Las interacciones reales se muestran con flechas en Figura - Modelo arquitectónico. Un modelo de capas más detallado se muestra en Figura - Modelo arquitectónico detallado.

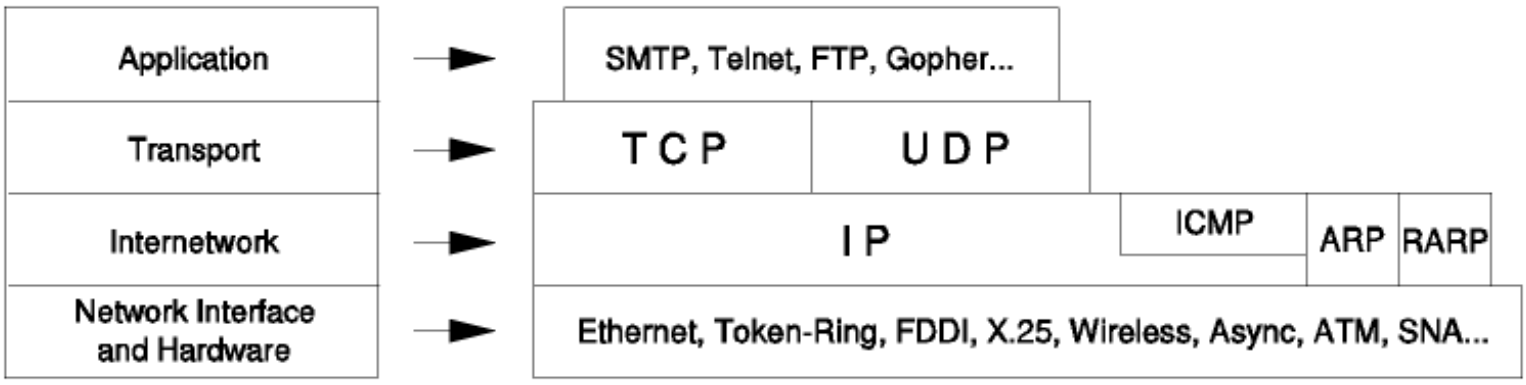


Figura: Modelo arquitectónico detallado

2.1.2.2 Puentes, "routers" y pasarelas

La formación de una red conectando múltiples redes se consigue por medio de los "routers". Es importante distinguir entre un "router", un puente y una pasarela.

Puente
: Interconecta segmentos de LAN a nivel de interfaz de red y envía tramas entre ellos. Un puente realiza la función de retransmisión MAC, y es independiente de cualquier capa superior (incluyendo el enlace lógico). Proporciona, si se necesita, conversión de protocolo a nivel MAC.
Un puente es transparente para IP. Es decir, cuando un host envía un datagrama a otro host en una red con el que se conecta a través de un puente, envía el datagrama al host y el dar cruza el puente sin que el emisor se dé cuenta.

"Router"
: Interconecta redes en el nivel de red y encamina paquetes entre ellas. Debe comprender la estructura de direccionamiento asociada con los protocolos que soporta y tomar la decisión de si se han de enviar, y cómo se ha de hacer, los paquetes. Los "routers" son capaces de elegir las mejores rutas de transmisión así como tamaños óptimos para los paquetes. La función básica de encaminamiento está implementada en la capa IP. Por lo tanto, cualquier estación de trabajo que ejecute TCP/IP se puede usar como "router".
Un "router" es visible para IP. Es decir, cuando un host envía un dar IP a otro host en una red conectada por un "router", envía el datagrama al "router" y no directamente al host de destino.

Pasarela

Interconecta redes a niveles superiores que los puentes y los "routers". Una pasarela suele soportar el mapeado de direcciones de una red a otra, así como la transformación de datos entre distintos entornos para conseguir conectividad entre los extremos de la comunicación. Las pasarelas limitan típicamente la conectividad de dos redes a un subconjunto de los protocolos de aplicación soportados en cada una de ellas.
Una pasarela es *opaca* para IP. Es decir, un host no puede enviar un datagrama IP a través de una pasarela: sólo puede enviarlo a la pasarela. La pasarela se ocupa de transmitirlo a la otra red con la información de los protocolos de alto nivel que vaya en él.

Estrechamente ligado al concepto de pasarela, está el de cortafuegos("firewall") o pasarela cortafuegos, que se usa para restringir el acceso desde Internet a una red o un grupo de ellas, controladas por una organización, por motivos de seguridad. Ver Cortafuegos para más detalles.

2.1.2.3 Encaminamiento IP

Los datagramas entrantes se chequean para ver si el host local es el destinatario:

sí
: El datagrama se pasa a los protocolos de nivel superior.

no
: El datagrama es para un host diferente. La acción depende del valor del flag "ipforwarding"(retransmisión IP).

verdadero
: El datagrama se trata como si fuera un datagrama saliente y se encamina el siguiente salto según el algoritmo descrito abajo.

falso
: El datagrama se desecha.

En el protocolo de red, los datagramas salientes se someten al algoritmo de encaminamiento IP que determina dónde enviar el datagrama de acuerdo con la dirección de destino.

- Si el host tiene una entrada en su tabla de encaminamiento IP (verEncaminamiento IP básico) que concuerde con la ir de destino, el datagrama se envía a la dirección correspondiente a esa entrada.
- Si el número de red de la dirección IP de destino es el mismo que el de uno de los adaptadores de red del host(están en la misma red) el datagrama se envía a la dirección física del host que tenga la dirección de destino.
- En otro caso, el datagrama se envía a un "router" por defecto.

Este algoritmo básico, necesario en toda implementación de IP, es suficiente para realizar las funciones de encaminamiento elementales.

Como se señaló arriba, un host TCP/IP tiene una funcionalidad básica como "router", incluida en IP. Un "router" de esta clase es adecuado para encaminamiento simple, pero no para redes complejas. Los protocolos requeridos para redes complejas se describen en protocolos de encaminamiento.

El mecanismo de encaminamiento IP, combinado con el modelo por capas de TCP/IP, se representa en Figura - El "router". Muestra un datagrama IP, yendo de una dirección IP(número de red X, host número A) a otra(número de red Y, host número B), a través de dos redes físicas. Nótese que en el "router" intermedio, sólo están implicados los niveles inferiores de la pila(red e interfaz de red).

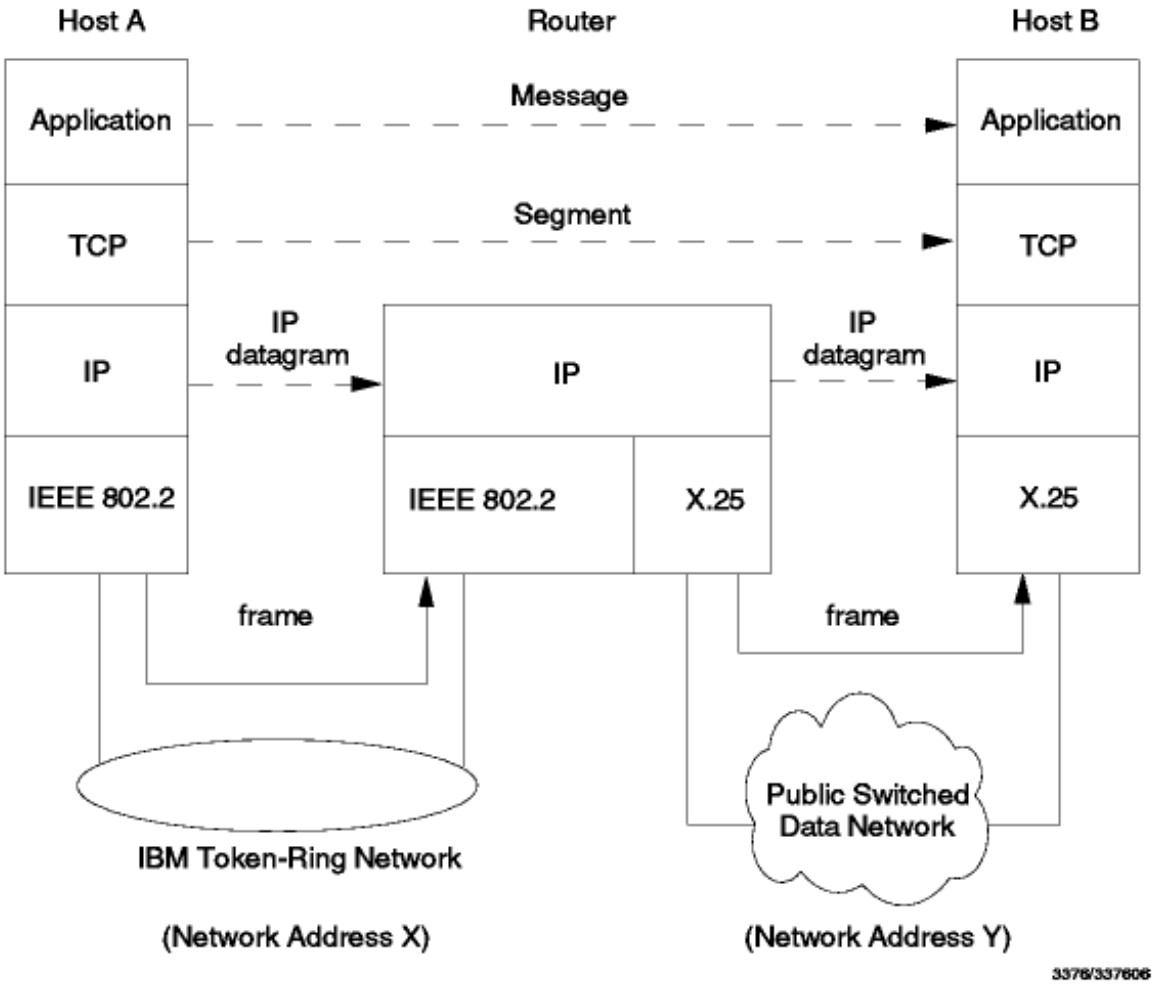


Figura: El "router" - La función de "router" la realiza el protocolo IP.

 Tabla de contenidos  Direccionamiento

2.2 Direccionamiento

Las direcciones de Internet pueden ser simbólicas o numéricas. La forma simbólica es más fácil de leer, por ejemplo: minombre@tcpip.com. La forma numérica es un número binario sin signo de 32 bits, habitualmente expresado en forma de números decimales separados por puntos. Por ejemplo, 9.167.5.8 es una dirección de Internet válida. La forma numérica es usada por el software de IP. La función de mapeo entre los dos la realiza el *DNS(Domain Name System)*discutido inDNS(Domain Name System). Primeramente examinaremos la forma numérica, denominada dirección IP.

2.2.1 La dirección IP

Los estándares para las direcciones IP se describen en *RFC 1166 -- Números de Internet*.

Para ser capaz de identificar una máquina en Internet, a cada interfaz de red de la máquina o host se le asigna una dirección, la *dirección IP*, o *dirección de Internet*. Cuando la máquina está conectada a más de una red se le denomina *"multi-homed"* y tendrá una dirección IP por cada interfaz de red. La dirección IP consiste en un par de números:

```
IP dirección = <número de red<número de interfaz de red
```

La parte de la dirección IP correspondiente al *número de red* está administrada centralmente por el InterNIC(Internet Network Information Center)y es única en toda la Internet.(1)

Las direcciones IP son números de 32 bits representados habitualmente *en formato decimal* (la representación decimal de cuatro valores binarios de 8 bits concatenados por puntos). Por ejemplo*128.2.7.9* es una dirección IP, donde 128.2 es el número de red y 7.9 el de la interfaz de red. Las reglas usadas para dividir una dirección de IP en su parte de red y de interfaz de red se explican abajo.

El formato binario para la dirección IP 128.2.7.9 es:

```
        10000000 00000010 00000111 00001001
```

Las direcciones IP son usadas por el protocolo IP(ver Internet Protocol (IP)) para definir únicamente un host en la red. Los datagramas IP(los paquetes de datos elementales intercambiados entre máquinas) se transmiten a través de alguna red física conectada a la interfaz de la máquina y cada uno de ellos contiene la *dirección IP de origen* y *la dirección IP de destino*. Para enviar un datagrama a una dirección IP de destino determinada la dirección de destino de ser traducida o mapeada a una dirección física. Esto puede requerir transmisiones en la red para encontrar la dirección física de destino(por ejemplo, en LANs el ARP("Adress Resolution Protocol", analizado en ARP("Address Resolution Protocol"), se usa para traducir las direcciones IP a direcciones físicas MAC).

Los primeros bits de las direcciones IP especifican como el resto de las direcciones deberían separarse en sus partes de red y de interfaz.

Los términos *dirección de red* y *netID* se usan a veces en vez de número de red, pero el término formal, utilizado en RFC 1166, es número de red. Análogamente, los términos *dirección de host* y *hostID* se usan ocasionalmente en vez de número de host.

Hay cinco clases de direcciones IP. Se muestran en Figura - Clases asignadas de direcciones de Internet.

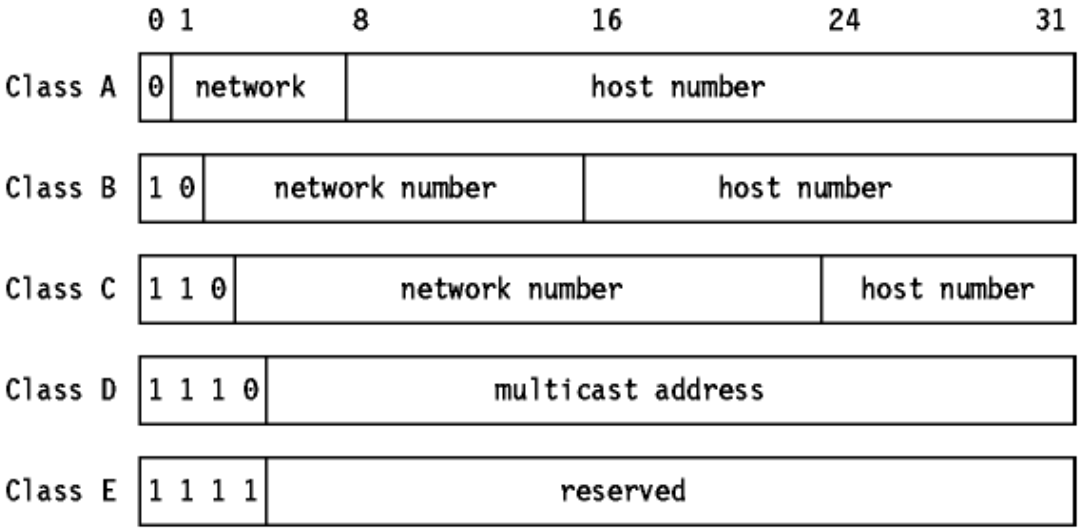


Figura - Clases asignadas de direcciones de Internet.

Nota: Dos de los números de red de cada una de las clases A, B y C, y dos de los números de host de cada red están preasignados: los que tienen todos los bits a 0 y los que tienen todos los bits a 1. Son estudiados más abajo en Direcciones IP especiales.

- Las direcciones de clase A usan 7 bits para el número de red permitiendo 126 posibles redes(veremos posteriormente que de cada par de direcciones de red y de host, dos tienen un significado especial). Los restantes 24 bits se emplean para el número de host, de modo que cada red tener hasta 16,777,214 hosts.
- Las direcciones de clase B usan 14 bits para el número de red, y 16 bits para el de host, lo que supone 16382 redes de hasta 65534 hosts cada una.
- Las direcciones de clase C usan 21 bits para el número de red y 8 para el de host, lo que supone 2,097,150 redes de hasta 254 hosts cada una.
- Las direcciones de clase D se reservan para multicasting o multidifusión, usada para direccionar grupos de hosts en un área limitada.VerMulticasting para más información sobre el multicasting.
- Las direcciones de clase E se reservan para usos en el futuro

Es obvio que una dirección de clase A sólo se asignará a redes con un elevado número de hosts, y que las direcciones de clase C son adecuadas para redes con pocos hosts. Sin embargo, esto significa que las redes de tamaño medio(aquellas con más de 254 hosts o en las que se espera que en el futuro haya más de 254 hosts) deben usar direcciones de clase IP. El número de redes de tamaño pequeño y medio ha ido creciendo muy rápidamente en los últimos años y se temía que, de haber permitido que se mantuviera este crecimiento, todas las direcciones de clase B se habrían usado para mediados de los '90. Esto es lo que se conoce como el problema del agotamiento de las direcciones IP. Este problema y cómo está siendo tratado es analizado en El problema del agotamiento de las direcciones IP.

Un hecho a señalar en la división de la dirección IP en dos partes es que esta división a su vez divide en dos partes la responsabilidad de elegir una dirección IP. El número de red es asignado por el InterNIC y el de host por la autoridad que controla la red. Como veremos en la siguiente sección, el número de host puede dividirse aún más: esta división también es controlada por la autoridad propietaria de la red, y *no* por el InterNIC.

2.2.2 Subredes

Debido al crecimiento explosivo de Internet, el uso de direcciones IP asignadas se volvió demasiado rígido para permitir cambiar con facilidad la configuración de redes locales. Estos cambios podían ser necesarios cuando:

- Se instala una nueva red física.
- El crecimiento del número de hosts requiere dividir la red local en dos o más redes.

Para evitar tener que solicitar direcciones IP adicionales en estos casos, se introdujo el concepto de *subred*.

El número de host de la dirección IP se subdivide de nuevo en un número de red y uno de host. Esta segunda red se denomina *subred*. La red principal consiste ahora en un conjunto de subredes y la dirección IP se interpreta como

```
<número de red<número de subred<número de host
```

La combinación del número de subred y del host suele denominarse "dirección local" o parte local". La creación de subredes se implementa de forma que es transparente a redes remotas. Un host dentro de una red con subredes es consciente de la existencia de estas, pero un host de un red distinta no lo es; sigue considerando la parte local de la dirección IP como un número de host.

La división de la parte local de la dirección IP en números de subred y de host queda a libre elección del administrador local; cualquier serie de bits de la parte local se puede tomar para la subred requerida. La división se efectúa empleando una *máscara de subred* que es un número de 32 bits. Los bits a cero en esta máscara indican posiciones de bits correspondientes al número de host, y los que están a uno, posiciones de bits correspondientes al número de subred. Las posiciones de la máscara pertenecientes al número de red se ponen a uno pero no se usan. Al igual que las direcciones IP, las máscaras de red suelen expresarse en formato decimal.

El tratamiento especial de "todos los bits a cero" y "todos los bits a uno" se aplica a cada una de las tres partes de dirección IP con subredes del mismo modo que a una dirección IP que no las tiene. Ver Direcciones IP especiales. Por ejemplo, una red de clase B con subredes, que tiene un parte local de 16 bits, podría hacer uso de uno de los siguientes esquemas:

- El primer byte es el número de subred, el segundo el de host. Esto proporciona 254(256 menos dos, al estar los valores 0 y 255 reservados) posibles subredes, de 254 hosts cada una. La máscara de subred es 255.255.255.0.
- Los primeros 12 bits se usan para el número de subred, y los 4 últimos para el de host. Esto proporciona 4094 posibles subredes(4096 menos 2), pero sólo 14 host por subred. La máscara de subred es 255.25.255.240. Hay muchas otras posibilidades.

Mientras el administrados es totalmente libre de asignar la parte de subred a la dirección local de cualquier forma legal, el objetivo es asignar un *número* de bits al número de subred y el resto a la dirección local. Por tanto, es corriente usar un bloque de bits contiguos al comienzo de la parte local para el número de subred ya que así las direcciones son más legibles(esto es particularmente cierto cuando la subred ocupa 8 o 16 bits). Con este enfoque, cualquiera de las máscaras anteriores es buena, pero no máscaras como 255.255.252.252 o 255.255.255.15.

2.2.2.1 Tipos de "subnetting"

Hay dos tipos de "subnetting": estático y de longitud variable. El de longitud variable es el más flexible de los dos. El tipo de "subnetting" disponible depende del protocolo de encaminamiento en uso; el IP nativo sólo soporta "subnetting" estático, al igual que el ampliamente utilizado RIP. Sin embargo, la versión 2 del protocolo RIP soporta además "subnetting" de longitud variable. Para ver una descripción de RIP y RIP2, ir a RIP("Routing Information Protocol"). Protocolos de encaminamiento analiza los protocolos de encaminamiento en detalle.

"Subnetting" estático

El "subnetting" estático consiste en que todas las subredes de la red dividida empleen la misma máscara de red. Esto es simple de implementar y de fácil mantenimiento, pero implica el desperdicio de direcciones para redes pequeñas. Por ejemplo, una red de cuatro hosts que use una máscara de subred de 255.255.255.0 desperdicia 250 direcciones IP. Además, hace más difícil reorganizar la red con una máscara nueva. Hoy en día, casi todos los hosts y "routers" soportan "subnetting" estático.

"Subnetting" de longitud variable

Cuando se utiliza "subnetting" de longitud variable, las subredes que constituyen la red pueden hacer uso de diferentes máscaras de subred. Una subred pequeña con sólo unos pocos hosts necesita una máscara que permita acomodar sólo a esos hosts. Una subred con muchos puede requerir una máscara distinta para direccionar esa elevada cantidad de hosts. La posibilidad de asignar máscaras de subred de acuerdo a las necesidades individuales de cada subred ayuda a conservar las direcciones de red. Además, una subred se puede dividir en dos añadiendo un bit a la máscara. El resto de las subredes no se verán afectadas por el cambio. No todos los hosts y "routers" soportan "subnetting" de longitud variable.

Sólo se dispondrán redes del tamaño requerido y los problemas de encaminamiento se resolverán aislando las redes que soporten "subnetting" de longitud variable. Un host que no soporte este tipo de "subnetting" debería disponer de una ruta de encaminamiento a un "router" que sí lo haga.

Mezclando "subnetting" estático y de longitud variable

A primera vista, parece que la presencia de un host que sólo puede manejar "subnetting" estático impediría utilizar "subnetting" de longitud variable en cualquier punto de la red. Afortunadamente no es este el caso. Siempre que los "routers" entre las subredes que tengan distintas máscaras usen "subnetting" de longitud variable, los protocolos de encaminamiento son capaces de ocultar la diferencia entre máscaras de subred a cada host de una subred. Los hosts pueden seguir usando encaminamiento IP básico y desentenderse de las complejidades del "subnetting", que quedan a cargo de "routers" dedicados a tal efecto.

2.2.2.2 Ejemplo de "subnetting" estático

Asumamos que a nuestra red se le ha asignado el número de red IP de clase B 129.112. Tenemos que implementar múltiples redes físicas en nuestra red, y algunos de los "routers" que usaremos no admiten "subnetting" de longitud variable. Por tanto tendremos que elegir una máscara de subred para la totalidad de la red. Tenemos una dirección local de 16 bits para la red y debemos dividirla correctamente en dos partes. Por el momento, no preveremos tener más de 254 redes físicas, ni más de 254 hosts por red, de tal forma que una máscara de subred aceptable sería 255.255.255.0(que además tiene la ventaja de ser legible). Esta decisión debe tomarse cuidadosamente, ya que será difícil cambiarla posteriormente. Si el número de redes o de hosts crece por encima de nuestras previsiones, puede que tengamos que implementar "subnetting" de longitud variable para usar al máximo las 65534 direcciones locales de las que disponemos.

Figura - Una configuración de subred muestra un ejemplo de implementación con tres subredes.

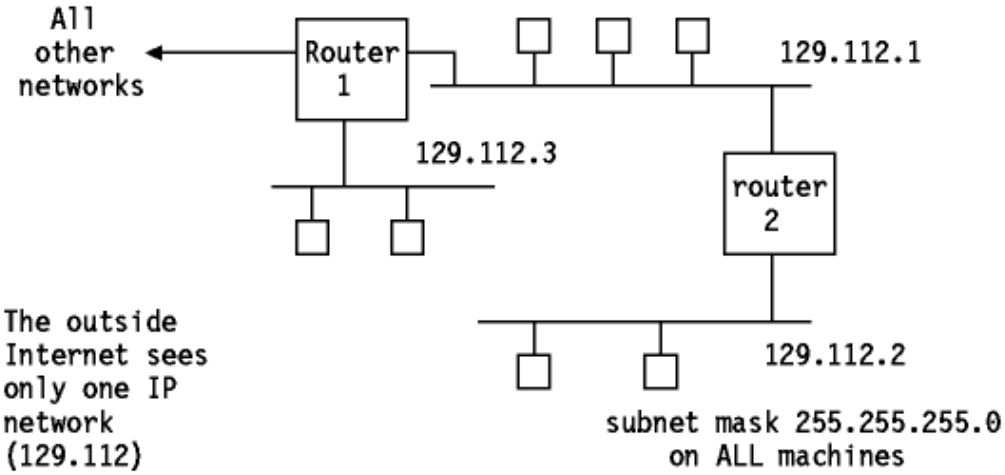
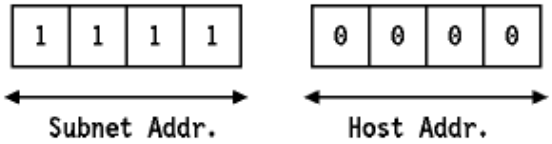


Figura: Una configuración de subred - Tres redes físicas forman una sola red IP. Los dos "routers" realizan tareas ligeramente diferentes. El "router" 1 actúa como "router" entre las subredes 1 y 3 así como para toda nuestra red y el resto de Internet. El "router" 2 actúa sólo como "router" entre las redes 1 y 2.

Consideremos ahora una máscara de subred diferente: 255.255.255.240. El cuarto octeto se ha dividido por tanto en dos partes:



La siguiente tabla contiene las posibles subredes que usarían esta máscara:

Hexadecimal value	Subnet number
0000	0
0001	16
0010	32
0011	48
0100	64
0101	80
0110	96
0111	112
1000	128
1001	144
1010	160
1011	176
1100	192
1101	208
1110	224
1111	240

Tabla: Valores de subredes para la máscara de subred 255.255.255.240

Para cada uno de estos valores de subred, sólo 14 direcciones(de la 1 a la 14) de hosts están disponibles, ya que sólo la parte derecha del octeto se puede usar y porque las direcciones 0 y 15 tienen un significado especial tal como se describe en Direcciones IP especiales.

De este modo, el número de subred 9.67.32.16 contendrá a los hosts cuyas direcciones IP estén en el rango de 9.67.32.17 a 9.67.32.30, y el número de subred 9.67.32.32 a los hosts cuyas direcciones IP estén en el rango de 9.67.32.33 a 9.67.32.46, etc.

2.2.2.3 Encaminamiento IP con subredes

Para encaminar un datagrama IP en la red, el algoritmo general de encaminamiento IP tiene la forma siguiente:

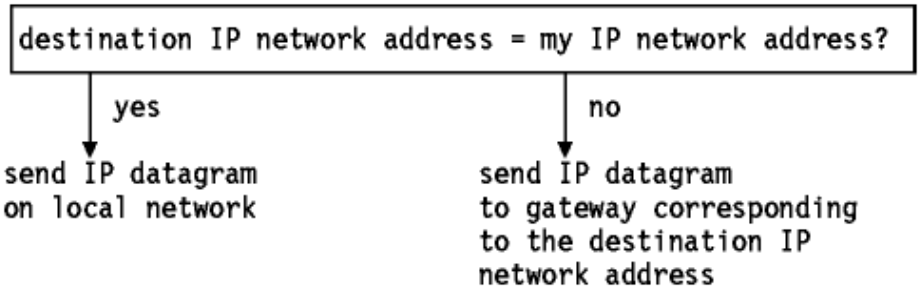


Figura: Encaminamiento IP con subredes

Para ser capaz de distinguir entre subredes, el algoritmo de encaminamiento IP cambia y adopta la siguiente forma:

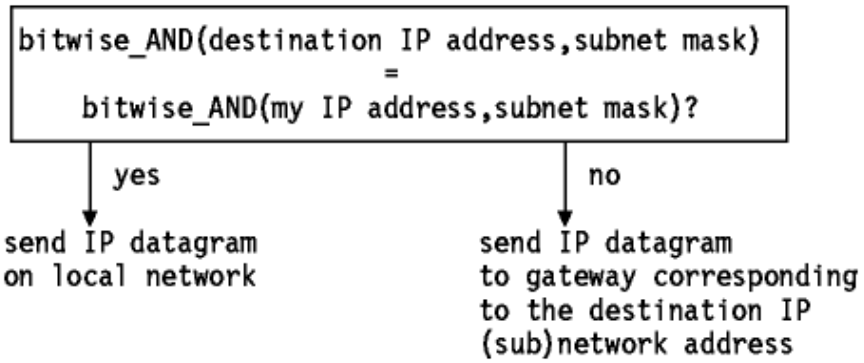


Figura: Encaminamiento IP con subredes

Algunas consecuencias de este algoritmo son:

- Es un cambio a algoritmo general. Por tanto, para poder operar de este modo, la correspondiente pasarela debe contener también el nuevo algoritmo. Algunas implementaciones pueden seguir usando el algoritmo general, y no funcionarán dentro de una red con subredes, aunque todavía podrán comunicarse con hosts en otras redes que no empleen "subnetting".
- Ya que el encaminamiento IP se usa en todos los hosts (aunque no en todos los "routers"), todos los hosts en la subred deben:
  1. Tener un algoritmo IP que soporte "subnetting".
  2. Tener la misma máscara de subred(a menos que existan subredes dentro de la subred).
- Si la implementación de algún host no soporta "subnetting", dicho host sólo podrá comunicarse con hosts de la propia subred, pero no con máquinas que se hallen en otra subred dentro de su misma red. Esto se debe a que el host sólo ve la red IP y su encaminamiento no puede distinguir entre un datagrama IP dirigido a un host de su subred y que se debería enviar a través de un "router" a una subred diferente.

En caso de que uno o más hosts no soporten "subnetting", una forma alternativa de lograr el mismo objetivo es hacer uso del *proxy-ARP*, que no requiere cambios al algoritmo de encaminamiento IP para un host con una sola interfaz("single-homed"), pero requiere cambios en los "routers" entre subredes. Esto se explica con más detalle en Proxy-ARP o "subnetting" transparente.

2.2.2.4 Obteniendo una máscara de subred

Habitualmente, los hosts almacenan su máscara de subred en un fichero de configuración. Sin embargo, a veces esto no se puede hacer, como es el caso de estaciones de trabajo sin disco. El protocolo ICMP incluye dos mensajes, solicitud de máscara de direcciones y respuesta de máscara de direcciones, que permitirá a los hosts obtener la máscara de subred correcta de un servidor. Ver Solicitud de máscara de direcciones(17) y Respuesta de máscara de direcciones (18) para más información.

2.2.2.5 Direccionando "routers" y hosts "multi-homed"

Un host se denomina *"multi-homed"* cuando tiene conexión física con múltiples redes o subredes . Todos los "routers" han de ser multi-homed ya que su trabajo es unir redes o subredes distintas. Un host multi-homed tiene siempre una dirección IP diferente para cada adaptador de red, puesto que cada adaptador se halla en una red distinta.

Hay una excepción aparente a esta regla: con algunos sistemas(por ejemplo VM y VMS) es posible especificar la misma dirección IP para múltiples enlaces punto a punto (como es el caso de los adaptadores de canal a canal) si el protocolo de encaminamiento se limita al algoritmo básico de encaminamiento IP.

2.2.3 Direcciones IP especiales

Como se ha señalado anteriormente, cualquier componente de un dirección IP con todos sus bits a 1 o a 0 tiene un significado especial

todos los bits a 0
: significa "este": "este" host (direcciones IP con <número de host=0) o "esta" red (direcciones IP con <número de red=0) y sólo se usa cuando el valor real no se conoce. Esta forma de expresar direcciones se utiliza con direcciones IP fuente, cuando el host trata de determinar sus direcciones IP por medio de un servidor remoto. El host puede incluir su número de host, si lo conoce, pero no su número de red o subred. Ver Protocolo BOOTstrap - BOOTP.

todos los bits a 1
: significa "todos": "todas" las redes o "todos" los hosts. Por ejemplo, 128.2.255.255 (una dirección de clase B con número de host 255.255) significar "todos los host de la red 128.2". Este forma de expresar direcciones se emplea en mensajes de broadcast, como se describe más abajo.

Hay otra dirección de especial importancia: el número de red de clase A con todos los bits a 1, 127, se reserva para la *dirección de loopback*. Todo lo que se envíe a una dirección con 127 como valor del byte de mayor orden, por ejemplo 127.0.0.1, no debe encaminarse a través de la red, sino directamente del controlador de salida al de entrada. (2)

2.2.4 Unicasting, broadcasting y multicasting

La mayoría de las direcciones IP se refieren a un sólo destinatario: se denomina direcciones de *unicast*. Sin embargo, como se ha señalado anteriormente, hay dos tipos especiales de direcciones IP que se utilizan para direccionar a múltiples destinatarios: las direcciones de broadcast y de multicast. Cualquier protocolo *no orientado a conexión* puede enviar

mensajes de broadcast o de multicast, además de los unicast. Un protocolo *orientado a conexión* sólo puede usar direcciones de unicast porque la conexión existe entre un par específico de hosts. Ver TCP("Transmission Control Protocol") para más información sobre los protocolos orientados a conexión..

2.2.4.1 Broadcasting

Hay una serie de direcciones que usan para el broadcast en IP: todas manejan el convenio de que "todos los bits a 1" indica "todos". Las direcciones de broadcast nunca son válidas como direcciones fuente, sólo como direcciones de destino. Los diferentes tipos de broadcast se listan aquí:

direcciones de broadcast limitado
: La dirección 255.255.255.255 (todos los bits a 1 en toda la dirección IP) se usa en redes que soportan broadcast, como por ejemplo redes en anillo, y se refiere a todos los host de la subred. No requiere que el host tenga conocimiento alguno de la configuración IP. Todos los host de la red local reconocerán la dirección, pero los "router" nunca enviarán el mensaje. Esta regla tiene una excepción, llamada *retransmisión BOOTP*. El protocolo BOOTP emplea el broadcast limitado para permitir a estaciones de trabajo sin disco contactar con un servidor BOOTP. La retransmisión BOOTP es una opción de configuración disponible en algunos "routers". Sin esta posibilidad, haría falta un servidor BOOTP en cada subred. Sin embargo, no se trata de una simple retransmisión, ya que el "router" también interviene en el desarrollo del protocolo BOOTP. Ver Protocolo BOOTstrap - BOOTP para más información al respecto.

direcciones de broadcast dirigidas a red
: Si el número de red es un válido, la red no se subdivide en subredes y el número de host referencia todos los hosts de la red especificada, (por ejemplo, 128.2.255.255). Los "router" deberían enviar estos mensajes de broadcast a menos que están configurados para no hacerlo. Este tipo de broadcast se utiliza en solicitudes ARP (ver ARP("Address Resolution")) en redes que contienen subredes.

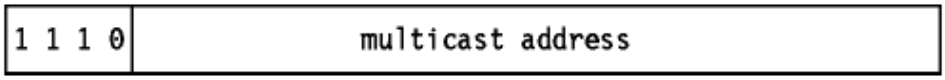
direcciones de broadcast dirigidas a subred
: Si el número de red y el de subred son válidos, y el de host tiene todos sus bits a 1, entonces la dirección referencia a todos los host de la subred especificada. Ya que la subred fuente y la de destino pueden tener distintas máscaras de subred, la fuente debe resolver de algún modo la máscara usada en la subred de destino. El broadcast lo efectúa realmente el "router" de subred que recibe el datagrama.

direcciones de broadcast dirigidas a todas las subredes
: Si el número de red es válido, la red se subdivide en subredes y la parte local de la dirección tiene todos los bits a 1(por ejemplo, 128.2.255.255), y la dirección se refiere a todos los hosts en todas las subredes de la red especificada. En principio, los "router" pueden propagar broadcasts por todas las subredes, aunque no están obligados a hacerlo. En la práctica, no lo hacen; hay pocas circunstancias en las que un broadcast sea deseable, y puede causar problemas, particularmente si un host se ha configurado incorrectamente sin su máscara de subred. Considerar el derroche de recursos que se produciría si el host 9.180.214.114 en la red local clase A con subredes no fuera consciente de la existencia de esas subredes y usara 9.255.255.255 como dirección de broadcast "local" en vez de 9.180.214.255 y todos los "router" aceptaran la solicitud de enviar mensajes a todos los clientes.

Si los "router" respetan todos los mensajes de broadcast dirigidos a subredes, utilizan un algoritmo llamado Retransmisión Inversa("*Reverse Path Forwarding")* para evitar que los mensajes de broadcast se multipliquen descontroladamente. Ver el RFC 922 para más detalles sobre este algoritmo.

2.2.4.2 Multicasting

El broadcast tiene una gran desventaja: su falta de selectividad. Si un datagrama IP se difunde por broadcast a una subred, cada host de la misma lo recibirá, y tendrá que procesarlo para determinar si el destinatario está activo. Si no lo está, el datagrama IP se elimina. El multicast elimina este overhead al usar grupos de direcciones IP.
Cada grupo está representado por un número de 28 bits, incluido en una dirección de clase D. Recordar que una dirección de clase D tiene el formato:



De este modo, *las direcciones de grupos de multicast* 224.0.0.0 a 239.255.255.255. Para cada dirección multicast hay un conjunto de cero o más hosts a la escucha. Es lo que se denomina el grupo de hosts. Para que un host envíe un mensaje a ese grupo no se requiere que pertenezca a él.

Hay dos clases de grupos de hosts:

permanentes
: La dirección IP tiene una asignación permanente a través de IANA. La pertenencia a un grupo no es permanente: un host puede unirse a un grupo o dejarlo a voluntad. Los grupos asignados con carácter permanente se incluyen en *STD 2 - Números asignados de Internet*. Algunos importantes son:

224.0.0.0
: Dirección base reservada

224.0.0.1
: Todos los sistemas en esta subred

224.0.0.2
: Todos los "routers" en esta subred

Algunos otros ejemplos usados por el protocolo de encaminamiento OSPF(ver Versión 2 de OSPF("Open Shortest Path First Protocol")) son:

224.0.0.5
: Todos los "router" OSPF

224.0.0.6
: "Routers" OSPF designados

Una aplicación puede además determinar la dirección IP permanente de un grupo por medio del DNS (ver DNS("Domain Name System")) usando el dominio mcast.net, o determinar el grupo permanente para una dirección a través de una consulta por punteros(ver Mapeando direcciones IP a nombres de dominio - Consultas por punteros) en el dominio 224.in-addr.arpa. Un grupo permanente existe aunque no tenga miembros.

provisionales
: Cualquier grupo que no sea permanente es provisional y está disponible para ser asignado dinámicamente según las necesidades. Los grupos provisionales dejan de existir cuando el número de sus miembros se hace cero.

El multicast en una sola red física que lo soporte s simple. Para unirse a un grupo, un proceso activo en un host debe informar de algún modo a sus controladores de red que desea ser parte del grupo especificado. El propio software de los controladores debe mapear la dirección de multicast a una dirección física de multicast para permitir la recepción de paquetes en esa dirección. Además, tiene que asegurarse de que el proceso receptor no recibe datagramas espúreos, chequeando la dirección de destino de la cabecera IP antes de pasarlos a la capa IP.

Por ejemplo, Ethernet soporta multicast si el byte de orden superior de la dirección de 48 bytes es X'01' y además IANA posee un bloque de la dirección, consistente en las direcciones entre X'00005E000000' y X'00005EFFFFFF'. IANA ha asignado la mitad inferior de este rango para direcciones de multicast, de modo que en una LAN Ethernet hay un rango de

direcciones físicas entre X'01005E000000' y X'01005E7FFFFF' usado para el multicast IP. Este rango tiene 23 bits utilizables, por lo que las direcciones de multicast de 28 bits se mapean a Ethernet tomando los 23 bits inferiores, es decir, hay 32 direcciones de multicast mapeadas sobre cada dirección Ethernet. Debido a este mapeo no unívoco, hace falta efectuar un filtrado en el controlador. Hay otras dos razones por la que se podría seguir necesitando el filtrado:

- Algunos adaptadores LAN están limitados a un número finito de direcciones multicast concurrentes y si este es excedido tendrán que recibir todos los multicast.
- Otros adaptadores LAN tienden a filtrar de acuerdo con un valor de una tabla de hash, lo que significa que hay una posibilidad de que el filtro tenga fugas, si dos direcciones multicast con el mismo valor de hash se usan al mismo tiempo.

A pesar de la necesidad de filtrar por software de paquetes multicast, el multicast aún causa mucho menos overhead en los hosts no interesados. En particular, aquellos hosts que no estén en ningún grupo no escuchan a los mensajes con direcciones multicast y por tanto todos los mensajes multicast son filtrados por el hardware de la interfaz de red.

El multicast no se limita a una sola red física. Hay dos aspectos del multicast en redes físicas a considerar:

- Un mecanismo para decidir la amplitud del multicast(recordar que a diferencia del unicast y el broadcast, las direcciones de multicast cubren toda Internet).
- Un mecanismo para decidir si un datagrama multicast necesita ser enviado a una red concreta.

El primer problema tiene fácil solución: el datagrama multicast tiene un TTL(tiempo de vida o "Time To Live") como cualquier otro datagrama, que se decrementa con cada salto a una nueva red Cuando el TTL se decrementa a cero, el datagrama no puede ir más lejos. El mecanismo para decidir si un router debe enviar un datagrama multicast se denomina *IGMP("Internet Group Management Protocol" o "Internet Group Multicast Protocol")*. IGMP se describe más detalle en IGMP("Internet Group Management Protocol"). IGMP y el multicast se definen en el *RFC 1112 - Extensiones de host para el multicast IP*.

2.2.5 El problema del agotamiento las direcciones IP

El número de redes en Internet se ha ido doblando aproximadamente cada año durante varios años. Sin embargo, el uso de las redes de clase A, B y C difiere mucho: la mayoría de las redes asignadas a finales de 1980 eran de clase B, y en 1990 se hizo evidente que, de continuar si la tendencia, el último número de red de clase B sería asignado en 1994. Por otro lado, las redes de clase C apenas se usaban.

La razón de esta tendencia era que la mayoría de los usuarios potenciales hallaban a las redes de clase B lo bastante grandes para sus necesidades previstas, ya que acomoda hasta 65534 hosts, mientras que un red de clase C, con un máximo de 254 hosts, restringe considerablemente el crecimiento potencial de hasta las redes pequeñas. Es más, la mayoría de las redes de clase B estaban asignadas a redes pequeñas. Hay un número relativamente pequeño de redes que necesiten 65,534 direcciones de hosts, pero muy pocas para las 254 sea un límite adecuado. En resumen, aunque las divisiones de clase A, B y C de las direcciones IP son lógicas y fáciles de usar(puesto que se producen a nivel de byte), en perspectiva no son las más prácticas, ya que las redes de clase C son demasiado pequeñas para la mayoría de las organizaciones mientras son demasiado grandes para ser bien aprovechadas por nadie excepto por las organizaciones más grandes.

Tabla - uso de las direcciones de red IP entre 1990 y 1994 muestra el uso de las direcciones de red IP entre 1990 y 1994.

Cls	Total	Year End 1990				Year End 1992				Year End 1994			
		Assigned		Allocated		Assigned		Allocated		Assigned		Allocated	
		Nbr	%	Nbr	%	Nbr	%	Nbr	%	Nbr	%	Nbr	%
A	126	38	30	101	80	51	40	114	90	53	42	116	92
B	16382	3238	20	4079	25	6812	42	7919	48	8432	51	9976	61
C	2097150	7792	0.4	104404	5.0	23339	1.1	200742	10	52833	2.5	521489	25

Tabla: uso de las direcciones de red IP entre 1990 y 1994. - Fuente: netinfo/ip_network_allocations.95Jan del FTP rs.internic.net

Algunos aspectos de esta tabla requieren explicación.

Assigned(asignado)
: La cantidad de números de red en uso. Las cantidades de la clase C son algo imprecisas, puesto que no incluyen muchas redes de clase C en Europa que se destinaron a RIPE, y fueron asignadas posteriormente, aunque aún están registradas como parte de RIPE.

Allocated(reservado)
: Incluye todas las redes asignadas y adicionalmente, aquellas redes que han sido reservadas bien por IANA(por ejemplo, todas las 63 redes de clase A) o que IANA ha sido destinado a registros nacionales que posteriormente podrán asignarlas. Por ejemplo, IANA reservó 64,783 redes de clase C en agosto de 1992, y 65,959 en julio de 1993.

Nota: Según IANA, el estado de un red es asignado *o* reservado, pero esta tabla trata el estado reservado como un superconjunto del asignado; el porcentaje real se puede calcular restando de 100 el porcentaje reservado para determinar cuánto espacio "libre" queda.

Otra forma de ver estos datos es examinar la proporción del espacio de direcciones que ha sido usado: las cantidades de la tabla no muestran, por ejemplo, que el espacio de direcciones de clase A es tan grande como el de las clases B y C combinados, o que teóricamente una sola red de clase A puede tener tantos host como 66,000 redes de clase C. Figura - Uso del espacio de direcciones IP muestra el uso del espacio de direcciones desde este punto de vista. El gráfico representa un espacio de direcciones de 32 bits, es decir, 4,294,967,296 direcciones. Las direcciones de clase A, B y C se dividen del modo siguiente:

Assigned(asignado)
: La porción del espacio de direcciones localizadas en redes asignadas. La cantidad real es en realidad mucho menor, puesto que es probable que cada red tenga bastante espacio libre, pero como este espacio no se puede utilizar fuera de la organización que controla la red, debe considerarse como espacio efectivo.

Se muestra en porciones resaltadas: el área resultante de unir todas las porciones representa la proporción del espacio de direcciones IP en uso.

Allocated(reservado)
: La porción del espacio de direcciones que se halla en redes que reservadas pero no asignadas más la porción de espacio perdida en números como el 0 de las redes de clase A y el 127(loopback).

El espacio reservado se muestra con porciones sombreadas.

Unallocated(no reservado)
: El resto del espacio de las clases A, B, C está libre; se muestra con una porción sin sombrear. Las porciones de las clases A, B, C se muestran con bordes progresivamente más finos. La clase A comienza a las 3 en punto y se mueve en sentido anti-horario hacia las clases B y C.

Class D(clase D)
: Una decimosexta parte del espacio total es absorbido por las direcciones de multicast de clase D. Se consideran direcciones en uso por lo que la porción correspondiente

aparece resaltada.

Class E(clase E)

La decimosexta parte restante del espacio de direcciones: IANA ha reservado esa parte, correspondiente a las direcciones IP con los cuatro bits de orden superior puestos a uno.

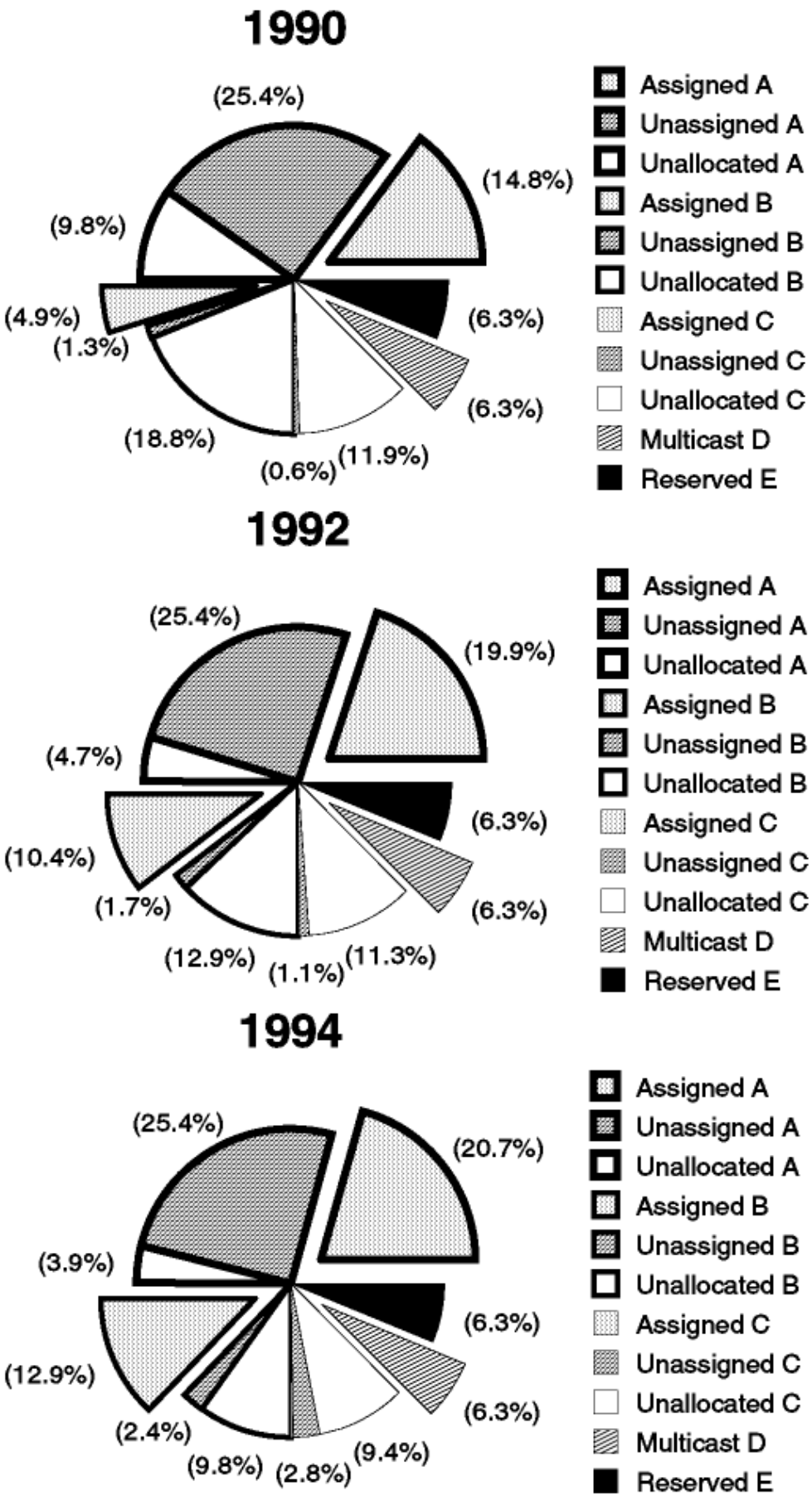


Figura: Uso del espacio de direcciones

Si se examina Tabla - Uso de las direcciones IP entre 1990 y 1994 se verá que es de 1990, el número de redes asignadas de clase B se ha ido incrementado a una tasa mucho más

baja que el número total de redes asignadas y que el agotamiento previsto de los números de clase B. La razón es que la política del InterNIC sobre la concesión de números de red cambió en desde 1990 para preservar el espacio de direcciones existente, en particular para frenar el agotamiento del espacio de direcciones de clase B. Las nuevas políticas se pueden resumir en:

- La mitad superior del espacio de direcciones de clase A se reserva indefinidamente para tener la posibilidad de usarlo en la transición a un nuevo sistema de numeración.
- Las redes de clase B sólo se asignan a organizaciones que puedan probar claramente que las necesitan. Lo mismo ocurre, por supuesto, con las direcciones de clase A. Los requerimientos para las redes de clase B son que la organización solicitante
    - Tenga un esquema de subnetting con más de 32 subredes dentro de su red operativa

      y

    - Tenga más de 4096 hosts

      Cualquier solicitud de una red de clase A se trataría considerando estrictamente el caso particular.

- A las organizaciones que no satisfacen los requerimientos para una red de clase B se les asigna un bloque de redes e clase C numeradas consecutivamente.
- La mitad inferior del espacio de direcciones de clase C(números de red del 192.0.0 al 223.255.245) se divide en 8 bloques que se para las autoridades regionales están reservadas del siguiente modo:

  192.0.0 - 193.255.255
  Multi-regional
  194.0.0 - 195.255.255
  Europa
  196.0.0 - 197.255.255
  Otros
  198.0.0 - 199.255.255
  Norte América
  200.0.0 - 201.255.255
  Centro y Sur América
  202.0.0 - 203.255.255
  Borde del Pacífico
  204.0.0 - 205.255.255
  Otros
  206.0.0 - 207.255.255
  Otros

  Los rangos definidos como "otros" se utilizan donde hace falta flexibilidad por encima de las limitaciones de las fronteras regionales. El rango definido como "multi-regional" incluye las redes de clase C que habían sido asignadas antes de que se adoptase este nuevo esquema. El InterNIC asignó 192 redes, y 193 habían sido previamente reservadas para el RIPE en Europa.

  La mitad superior del espacio de direcciones de clase C(208.0.0 a 223.255.255) permanece sin asignar y sin reservar.

- En las organizaciones que tienen una serie de números de clase C, el rango asignado contiene números de red *contiguos a nivel de bit* y el número de redes de ese rango es una potencia de dos. Es decir, todas las direcciones IP en ese rango tienen un prefijo común, y cada dirección con ese prefijo está a su vez dentro del rango. Por ejemplo, a una organización europea que requiera 1500 direcciones IP se le asignarían 8 números de red de clase C(2048 direcciones IP) del espacio reservado para redes europeas (194.0.0 a 195.255.255) y el primero de estos números de red sería divisible por ocho. Un rango de direcciones que se adecuase a esta regla sería el 194.32.136 - 194.32.143, en cuyo caso contendría todas las direcciones IP con el prefijo de 21 bits 194.32.136, o '11000010001000000010001'.

  El número máximo de números de red asignados contiguamente es 64, correspondiente a un prefijo de 18 bits. Una organización que requiera más de 4096 direcciones pero menos de 16,384 puede solicitar tanto una clase B como un rango de direcciones de clase C. En general, el número de clases C asignadas es el mínimo necesario para proporcionar la cantidad requerida de direcciones IP a la organización partiendo de una previsión sobre el plazo de los dos años siguientes. Sin embargo, en algunos casos, una organización puede solicitar múltiples redes que sean tratadas por separado. Por ejemplo, a una organización con 600 hosts se le asignarían normalmente 4 redes de clase C. No obstante, si esos hosts estuvieran distribuidos a lo largo de 10 LANs en anillo con testigo con entre 50 y 70 hosts por LAN, tal esquema de direcciones causaría graves problemas, ya que la organización tendría que encontrar 10 subredes dentro de un rango de direcciones locales de 10 bits. Esto significaría que al menos alguna de las LAN tendría una máscara de subred de 255.255.255.192, que sólo permite 62 hosts por LAN. La intención de las reglas no es forzar a la organización a que tenga un complicado sistema de subredes, así que la organización debería solicitar 10 números de clase C diferentes, uno para cada LAN.

Las reglas actuales se pueden encontrar en el *RFC 1466 - Directrices para la gestión del espacio de direcciones IP*. Las razones de las reglas de distribución de los números de red de clase C quedarán patentes en la próxima sección. Usar números de clase C de esta forma ha frenado el problema del agotamiento de las direcciones de clase B, pero no es una solución definitiva a las limitaciones de espacio inherentes a IP. La solución a largo plazo se discute en IP: La próxima generación(IPng).

2.2.6 Redes privadas

Otro enfoque de la conservación del espacio de direcciones IP se describe en el *RFC 1597 - Distribución de direcciones para redes privadas*. En pocas palabras, relaja la regla de que las direcciones IP han de ser unívocas globalmente al reservar parte del espacio de direcciones para redes que se usan exclusivamente dentro de una sola organización y que no requieren conectividad IP con Internet. Hay tres rangos de direcciones que IANA ha reservado con este propósito:

- *10* Una sola red de clase A
- 16 redes clase B contiguas del *172.16 al 172.31*
- 256 redes clase C contiguas del *192.168.0 al 192.168.255*

Cualquier organización puede usar cualquier dirección en estos rangos si no hace referencia a ninguna otra organización. Sin embargo, debido a que estas direcciones no son unívocas a nivel global, no pueden ser direccionadas por hosts de otras organizaciones y no están definidas para los "routers" externos. Se supone que los "routers" de una red que no usa direcciones privadas, particularmente aquellos operados por proveedores de servicios de Internet, han de desechar toda información de encaminamiento relativa a estas direcciones. Los "router" de una organización que utiliza direcciones privadas deberían limitar todas las referencias a direcciones privadas a los enlaces internos; no deberían hacer públicas las rutas a direcciones privadas ni enviar datagramas IP con estar direcciones a los "routers" externos. Los hosts que sólo tienen una dirección IP privada carecen de conexión IP con Internet. Esto puede ser deseable y a lo mejor puede ser una razón para emplear direcccionamiento privado. Toda la conectividad con host externos de Internet la deben proporcionar pasarelas de aplicación.

2.2.7 CIDR("Classless Inter-Domain Routing")

El uso de un rango de direcciones de clase C en vez de una sola de clase B acarrea un gran problema: cada red ha de ser direccionada por separado. El encaminamiento IP estándar sólo comprende las clases A, B y C. Dentro de cada uno de estos tipos de red, se puede usar "subnetting" para proporcionar mejor granularidad del espacio de direcciones en cada red, pero no hay forma de especificar que existe una relación real entre múltiples redes de clase C. El resultado de esto se denomina el problema de la *explosión de la tabla de encaminamiento*: una red de clase B de 3000 host requiere una entrada en la tabla de encaminamiento para cada "router" troncal, pero si la misma red se direccionase como un rango de redes de clase C, requeriría 16 entradas.

La solución a este problema es un método llamado *CIDR("Classless Inter-Domain Routing")*. El CIDR es un *protocolo propuesto como estándar* con status *electivo*.

El CIDR no encamina de acuerdo a la clase del número de red(de ahí el término "classless": sin clase) sino sólo según los bits de orden superior de la dirección IP, que se denominan *prefijo IP*. Cada entrada de encaminamiento CIDR contiene una dirección IP de 32 bits y una máscara de red de 32 bits, que en conjunto dan la longitud y valor del prefijo IP. Esto se puede representar como <dir_IP máscara_red. Por ejemplo, <194.0.0.0 254.0.0.0 representa el prefijo de 7 bits B'1100001'. CIDR maneja el encaminamiento para un grupo de redes con un prefijo común con una sola entrada de encaminamiento. Esta es la razón por la que múltiples números de red de clase C asignados a una sola organización tienen un prefijo común. Al proceso de combinar múltiples redes en una sola entrada se le llama *agregación de direcciones* o *reducción de direcciones*. También se le llama *supernetting* poque el encaminamiento se basa en máscaras de red más cortas que la máscara de red *natural* de la dirección IP, en contraste con el *subnetting*, donde las máscaras de red son más largas que la máscara natural.

A diferencia de las máscaras de subred, que normalmente son contiguas pero pueden tener una parte local no contigua, las máscaras de superred son *siempre* contiguas

Si se representan las direcciones IP con una árbol que muestre la topología de encaminamiento, donde cada hoja del árbol significa un grupo de redes que se consideran como una sola unidad(llamada *dominio de encaminamiento*) y el esquema de direccionamiento IP se elige de modo que cada bifurcación del árbol corresponda a un incremento en la longitud del prefijo IP, entonces el CIDR permite realizar la agregación de direcciones muy eficientemente. Por ejemplo, si un "router" en Norteamérica encamina todo el tráfico europeo a través de un único enlace, entonces una sola entrada de encaminamiento para <194.0.0.0 254.0.0.0 incluye el grupo de direcciones de redes de clase C asignadas a Europa como se describe más arriba. Esta única entrada toma el lugar de todas las entradas de los números de red asignados, que son un máximo de 2exp17, o 131,072. En el extremo europeo del enlace, hay entradas de encaminamiento con prefijos más largos que mapean la topología de la red europea, pero esta información de encaminamiento no hace falta en el extremo americano. La filosofía de CIDR es *"la mejor aproximación es la que tiene más aciertos"*, de modo que si los US necesitan hacer una excepción para un rango de direcciones, como por ejemplo las 64 redes <195.1.64.0 255.555.192.0, necesita sólo una entrada adicional, que en la tala de encaminamiento se superpone a otras entradas más generales(más cortas) de las redes que contiene. De este ejemplo se hace evidente que a medida que aumenta el uso del espacio de direcciones IP, particularmente de las de clase C, los beneficios de CIDR aumentan por igual, siempre que la asignación de direcciones siga la topología de la red. El estado actual del espacio de direcciones IP no sigue este esquema ya que el desarrollo de CIDR fue posterior. Sin embargo, se están asignando nuevas direcciones de clase C de tal modo que sean compatibles con CIDR, lo que debería tener el efecto de aliviar el problema de la explosión de las tablas de encaminamiento a corto plazo. A largo plazo, puede que sea necesaria una reestructuración del espacio de direcciones IP siguiendo pautas topológicas. Esto supondría tener que renumerar un gran número de redes, implicando un enorme trabajo de implementación, por lo que se trataría de un proceso gradual

Asumir que la topología de encaminamiento se puede representar con un simple árbol es un exceso de simplificación; aunque la mayoría de los dominios de encaminamiento tienen un sólo enlace que proporciona acceso al resto de Internet, hay también muchos dominios con enlaces múltiples. Los dominios de encaminamiento de estos dos tipos se llaman *"single-homed"(unipuerto)* y *"multi-homed"(multipuerto)*. Es más, la topología no es estática. No sólo se unen nuevas organizaciones a un ritmo creciente, sino que las ya existentes pueden cambiar partes de su topología, por ejemplo, si cambian de proveedor de servicios por razones comerciales o de otra índole. Aunque estos casos complican la implementación práctica del encaminamiento basado en CIDR y reducen la eficiencia de la agregación de direcciones que se puede conseguir, la estrategia no deja de ser válida.

Las políticas actuales para la distribución de direcciones de Internet y las suposiciones en las que se basan se describen en el *RFC 1518 - Una arquitectura para la distribución de direcciones IP con CIDR*. Se pueden resumir en:

- La asignación de direcciones IP refleja la topología física de la red y no de la organización; las restricciones organizacionales y administrativas no deberían usarse en la asignación de direcciones IP cuando *no* se ajusten a la topología de la red.
- En general, la topología de la red seguirá de cerca los límites continentales y nacionales, y por tanto las direcciones IP se deberían asignar partiendo de esta base.
- Habrá un número relativamente pequeño de redes que transportarán una elevada cantidad de tráfico entre dominios de encaminamiento y que estarán conectadas de modo no jerárquico, traspasando los límites nacionales. Estas redes se denominan *TRDst("Transit Routing Domains")*. Cada TRD tendrá un prefijo IP unívoco. En general, los TRDs no se organizarán jerárquicamente. Sin embargo, cuando un TRD se halle por completo dentro de los límites contientales, su prefijo IP debería ser una extensión del prefijo IP continental.
- Habrá organizaciones con enlaces a otras organizaciones que son para su uso privado y que no transportarán tráfico dirigido a otros dominios(tráfico de tránsito). Estas conexiones privadas no tienen un efecto significativo sobre la topología de red y pueden ser ignoradas.
- La gran mayoría de los dominios de encaminamiento serán single-homed. Es decir, estarán conectadas a un sólo TDR. Se les debería asignar direcciones que comiencen por el prefijo IP de ese TRD. Por tanto, todas las direcciones de los dominios "single-homed" conectados a un TDR se pueden agregar en una sola entrada de la tabla de encaminamiento para todos los dominios externos a ese TRD.

  Nota: Esto implica que si una organización cambia su proveedor de servicios de Internet, debería cambiar todas sus direcciones IP. Esto no es lo habitual, pero es probable que la difusión de CIDR lo convierta en una práctica mucho más común.

- Hay una serie de esquemas de asignación de direcciones que se pueden usar con dominios "multi-homed". Algunos son:
  - El uso de un único prefijo IP para el dominio. Los "routers" externos deben tener una entrada para la organización que se halla parcial o totalmente fuera de la jerarquía normal. Donde un dominio sea "multi-homed", pero todos los TRDs conectados estén topológicamente cerca, sería apropiado que el prefijo IP del dominio incluyese los bits comunes a todos los TRDs conectados. Por ejemplo, si todos los TRDs estuvieran totalmente dentro de los Estados Unidos, un prefijo IP indicando exclusivamente un dominio de Norteamérica sería lo adecuado.
  - El uso de un prefijo IP para cada TRD conectado, con hosts en el dominio que tengan direcciones IP que contengan el prefijo del TRD más apropiado. La organización da la impresión de ser un conjunto de dominios de encaminamiento.
  - Asignar un prefijo IP de uno de los TRDs conectados. Este TRD se convierte en un TRD por defecto para el dominio, aunque otros dominios pueden encaminar explícitamente sus mensajes por uno de los TRDs alternativos.
  - El uso de prefijos IP para referirse a conjuntos de dominios "multi-homed" con conexiones a TRDs. Por ejemplo, puede haber un prefijo IP que se refiera a dominios "single-homed" conectados a la red A, uno que se refiera a dominios "single-homed" conectados a la red B y uno para los dominios conectados a A y a B.
- Cada uno de estos esquemas tiene sus ventajas, desventajas y efectos colaterales. Por ejemplo, el primero tiende a generar un tráfico interno en el dominio receptor más cercano al host fuente superior al generado por el segundo esquema, aumentando recursos de red consumidos en la organización receptora.

  Debido a que los dominios "multi-homed" varían mucho en su carácter y ninguno de los esquemas anteriores parece apropiado para todos, no existe una política que sea la mejor, y el RFC 1518 no especifica ninguna regla para elegir entre ellas.

2.2.7.1 Implementación de CIDR

La implementación de CIDR en Internet se basa fundamentalmente en *BGP("Border Gateway Protocol", versión 4)* (ver BGP-4("Border Gateway Protocol", versión 4)). En el futuro CIDR también se implementará con una variante del estándar ISO *IDRP, ISO 10747("Inter-Domain Routing Protocol")*, llamado IDRP para IP, que está estrechamente relacionado con BGP-4.

La estrategia de implementación, descrita en el *RFC 1520 - Intercambiando información de encaminamiento a través de las fronteras de los proveedores en el entorno CIDR*, implica un proceso por fases a través de la jerarquía de encaminamiento, empezando por los "routers" troncales. Los proveedores de servicios de red se dividen en cuatro tipos:

Tipo 1
: Aquellos que no pueden emplear ningún IDRP.

Tipo 2
: Aquellos que usan IDRP por defecto pero que requieren rutas explícitas para una proporción considerable de los números IP de red asignados.

Tipo 3
: Aquellos que usan IDRP por defecto y añaden además un pequeño número de rutas explícitas.

Tipo 4
: Aquellos que ejecutan IDRP utilizando sólo rutas por defecto.

La implementación de CIDR implica una primera fase por medio de los proveedores de tipo 0, luego los de tipo 2 y finalmente los de tipo 3. CIDR ya se ha instituido ampliamente en troncales y más de 9000 rutas se han reemplazado por aproximadamente 2000 rutas CIDR.

2.2.7.2 Referencias

- *RFC 1467 - Difusión de CIDR en Internet*
- *RFC 1517 - Condiciones de aplicabilidad de CIDR*
- *RFC 1518 - Una arquitectura para la distribución de direcciones IP con CIDR*
- *RFC 1519 - CIDR: asignación de direcciones y estrategia de agregación*
- *RFC 1520 -Intercambiando información de encaminamiento a través de las fronteras de los proveedores en el entorno CIDR*

2.2.8 DNS("Domain Name System")

El protocolo DNS es un *protocolo estándar* (STD 13). Su status es *recomendado*. Es descrito en:

- *RFC 1034 - Nombres de dominio - conceptos y servicios*
- *RFC 1035 - Nombres de dominio - implementación y especificación*

Las configuraciones iniciales de Internet requerían que los usuarios emplearan sólo direcciones IP numéricas. Esto evolucionó hacia el uso de nombres de host simbólicos muy rápidamente. Por ejemplo, en vez de escribir TELNET 128.12.7.14, se podría escribir TELNET eduvm9, y eduvm9 se traduciría de alguna forma a la dirección IP 128.12.7.14. Esto introduce el problema de mantener la correspondencia entre direcciones IP y nombres de máquina de alto nivel de forma coordinada y centralizada.

Inicialmente, el NIC("Network Information Center") mantenía el mapeado de nombres a direcciones en un sólo fichero(HOSTS.TXT) que todos los hosts obtenían vía FTP. Se denominó *espacio de nombres plano*.

Debido al crecimiento explosivo del número de hosts, este mecanismo se volvió demasiado tosco(considerar el trabajo necesario sólo para añadir un host a Internet) y fue sustituido por un nuevo concepto: *DNS("Domain Name System")*. Los hosts pueden seguir usando un espacio de nombres local plano(el fichero HOSTS.LOCAL) en vez o además del DNS, pero fuera de redes pequeñas, el DNS es prácticamente esencial. El DNS permite que un programa ejecutándose en un host le haga a otro host el mapeo de un nombre simbólico de nivel superior a una dirección IP, sin que sea necesario que cada host tenga una base de datos completa de los nombres simbólicos y las direcciones IP.

En el resto de esta sección examinaremos cómo funciona el DNS desde el punto de vista del usuario. Ver DNS("Domain Name System") para más detalles sobre la implementación y los tipos de datos almacenados en DNS.

2.2.8.1 El espacio de nombres jerárquico

Consideremos la estructura interna de una gran organización. Como el jefe no lo puede hacer todo, la organización tendrá que partirse seguramente en divisiones, cada una de ellas autónoma dentro de ciertos límites. Específicamente, el ejecutivo a cargo de una división tiene autoridad para tomar decisiones sin requerir el permiso de su jefe.

Los nombres de dominio se forman de modo similar, y con frecuencia reflejarán la delegación jerárquica de autoridades usada para asignarlos. Por ejemplo, considerar el nombre

```
lcs.mit.edu
```

Aquí, lcs.mit.edu es el nombre de dominio de nivel inferior, un subdominio de mit.edu, que a su vez es un subdominio de edu("education"), conocido como *dominio raíz*. También podemos representar este forma de asignar nombres con un árbol jerárquico,(ver Figura - Espacio de nombres jerárquico).

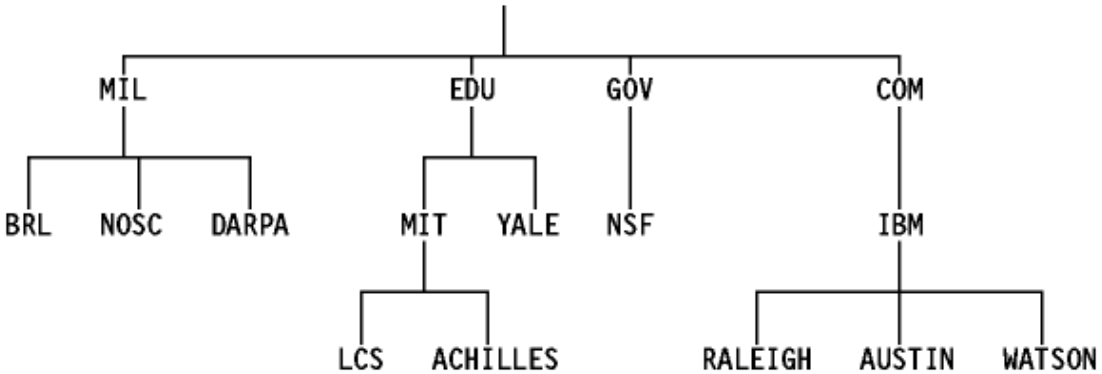


Figura: Espacio de nombres jerárquico - Esta figura muestra la cadena de autoridades en la asignación de nombres de dominio. Este árbol es sólo una fracción mínima del espacio de nombres real.

Figura - Los dominios genéricos de la cima muestra algunos de los dominios de la cima. El dominio único que se halla sobre la cima no tiene nombre y se le conoce como *dominio raíz*. La estructura completa se explica en las siguientes secciones.

2.2.8.2 FQDNs("Fully Qualified Domain Names")

Cuando se usa el DNS, es común trabajar con sólo una parte de la jerarquía de dominios, por ejemplo el dominio ral.ibm.com. El DNS proporciona un método sencillo para minimizar la cantidad de caracteres a escribir en estos casos. Si el nombre de dominio termina en un punto(por ejemplo wtscpok.itsc.pok.ibm.com.) se asume que está completo. Es lo que se llama un *FQDN("Fully Qualified Domain Name")* o *nombre absoluto de dominio*. Si, sin embargo, no termina en punto,(por ejemplo wtscpok.itsc) estará incompleto y procesador de nombres del DNS, como se verá más abajo, podrá completarlo, por ejemplo, añadiendo un sufijo como .pok.ibm.com al nombre de dominio. Las reglas para hacer esto dependen de la implementación y son configurables localmente.

2.2.8.3 Dominios genéricos

A los tres dominios de la cima se les llama dominios *genéricos* u *organizacionales*.

Nombre de dominio

Significado

edu
Instituciones educativas

gov
Instituciones gubernamentales

com
Organizaciones comerciales

mil
Grupos militares

net
Redes

int
Organizaciones internacionales

org
Otras organizaciones

Figura: Los dominios genéricos de la cima

Puesto que Internet comenzó en los Estados Unidos, la estructura del espacio de nombres jerárquico tenía inicialmente sólo organizaciones estadounidenses en la cima de la jerarquía, y sigue siendo cierto que gran parte de las organizaciones de la cima de la jerarquía son estadounidenses. Sin embargo, sólo los dominios .gov y .mil están restringidos a los US.

2.2.8.4 Dominios de países

Además hay dominios de nivel de cima para cada uno de los códigos internacionales de dos caracteres ISO 3166 para países(de *ae* para los Emiratos Árabes Unidos a *zw* para Zimbabwe). Se les conoce como dominios de *países* o dominios *geográficos*. Muchos países tienen sus propios dominios de segundo nivel por debajo, paralelamente a los dominios genéricos. Por ejemplo, en el Reino Unido, los dominios equivalentes a .com y .edu son .co.uk y .ac.uk("ac" es la abreviatura de "academic"). Está también el dominio .us, organizado geográficamente por estados(por ejemplo, .ny.us se refiere al estado de New York). Ver el RFC 1480 para una descripción detallada del dominio .us.

2.2.8.5 Mapeando nombres de dominio a direcciones IP

El mapeado de nombres a direcciones, proceso denominado resolución de nombres de dominio, lo proporcionan sistemas independientes cooperativos, llamados *servidores de nombres*. Un servidor de nombres es un programa servidor que responde a peticiones de un cliente llamado *procesador de nombres*.

Cada procesador de nombres está configurado con el nombre del servidor que va a usar(y posiblemente una lista de servidores alternativos con los que contactar si el servidor primario no está disponible). Figura - Resolución de nombres de dominio muestra esquemáticamente cómo un programa utiliza un procesador de nombres para convertir el nombre de un host en una dirección IP. El usuario proporciona el nombre de un host, y al programa de usuario emplea una rutina de librería, llamada stub, para comunicarse con un servidor de nombres que resuelve el nombre del host en una dirección IP y se la devuelve al stub, que a su vez lo devuelve al programa principal. El servidor de nombres pude obtener la respuesta de su caché de nombres, su propia base de datos o cualquier otro servidor de nombres.

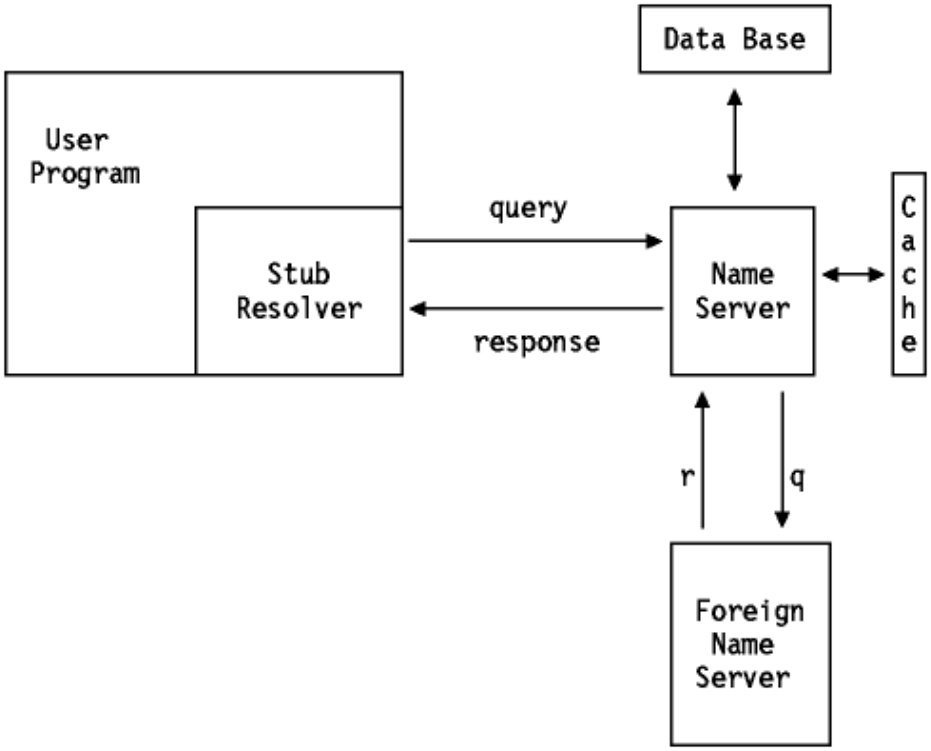


Figura: Resolución de nombres de dominio

2.2.8.6 Mapeando direcciones IP a nombres de dominio - Consultas con punteros

El DNS suministra el mapeado de nombres simbólicos a direcciones IP *y vice versa.* Mientras que en principio es algo sencillo buscar en la base de datos una dirección IP, dado su nombre simbólico, el proceso inverso no se puede hacer respetando la jerarquía. Por este motivo, existe otro espacio de nombres para el mapeado inverso. Se halla en el dominio in-addr.arpa ("arpa" porque Internet era originalmente la red de ARPA). Como las direcciones IP suelen escribirse en formato decimal con puntos, hay una capa de dominios para cada jerarquía. Sin embargo, debido a que los nombres de dominio tienen primero la parte menos significativa del nombre y el formato decimal con puntos los bytes más significativos primero, la dirección decimal se muestra en orden inverso. Por ejemplo, el dominio del DNS correspondiente a la dirección IP 129.34.139.30 es 30.139.34.129.in-addr.arpa. Dada una dirección IP, el DNS puede utilizarse para encontrar el nombre del host que sea su pareja. Una consulta de nombre de dominio para encontrar los nombres del host asociado a una dirección IP se llama "consulta con puntero".

2.2.8.7 Otros usos para el DNS

EL DNS está designado para ser capaz de almacenar una gran cantidad de información. Una de las más importantes es información del *intercambio de correo*, usada para el encaminamiento del correo electrónico. Esto aporta dos servicios: transparencia al reencaminar el correo a un host distinto del especificado y la implementación de pasarelas de correo, que pueden recibir correo electrónico y redirigirlo usando un protocolo diferente de aquel con el que lo reciben. Para más detalles, remitirse a SMTP y el DNS.

2.2.8.8 Referencias

Para más detalles sobre la implementación del DNS y el formato de mensajes entre servidores de nombres, ver DNS("Domain Name System"). Los siguientes RFCs definen el estándar de DNS y la información que almacena.

- *RFC 1032 - Guía de administrador de DNS*
- *RFC 1033 - Guía de las operaciones de administrador de DNS*
- *RFC 1034 - Nombres de dominio - Conceptos y servicios*
- *RFC 1035 - Nombres de dominio - Implementación y especificación*
- *RFC 1101 - Codificación DNS de nombres de red y de otros tipos*
- *RFC 1183 - Nuevas definiciones del DNS RR*
- *RFC 1706 - Registros de recursos DNS NSAP*

Tabla de contenidos IP ("Internet Protocol")

2.3 IP("Internet Protocol")

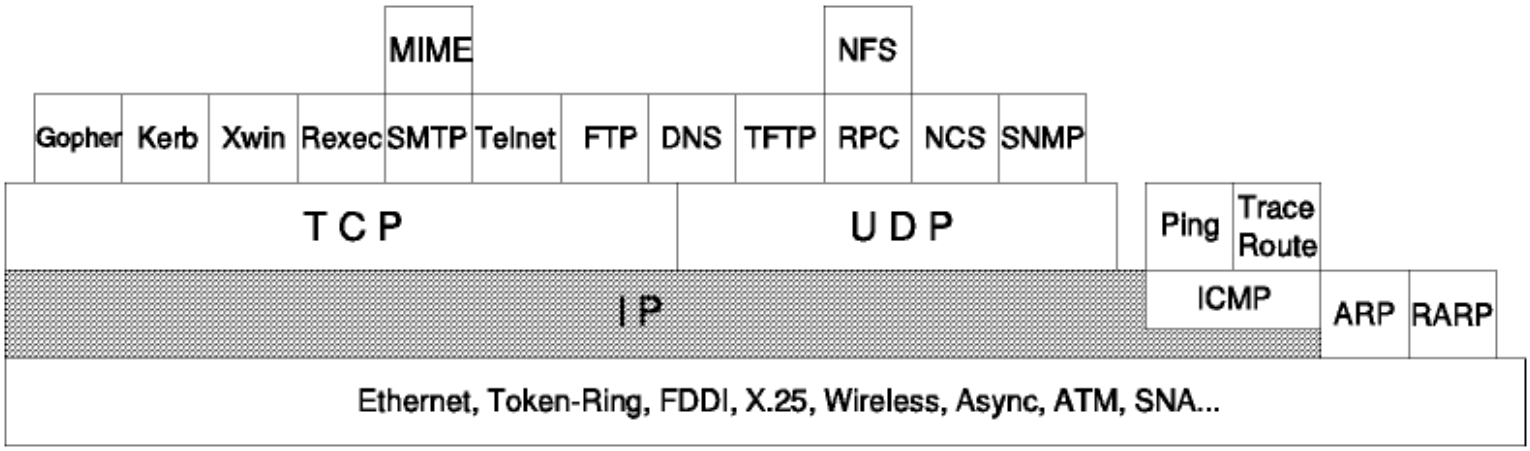


Figura: IP("Internet Protocol")

IP es un *protocolo estándar* con STD 5 que además incluye ICMP (ver ICMP("Internet Control Message Protocol")) e IGMP (ver IGMP("Internet Group Management Protocol")). Su status es *requerido*.

Su especificación actual se puede encontrar en los RFCs 791, 950, 919 y 922, que actualizado en el RFC 1349.

IP es el protocolo que oculta la red física subyacente creando una vista de *red virtual.* Es un protocolo de entrega de paquetes no fiable y no orientado a conexión, y se puede decir que aplica la ley del mínimo esfuerzo.

No aporta fiabilidad, contrtol de flujo o recuperación de errores a los prots de red inferiores. Los paquetes (*datagramas*) que envía IP se pueden perder, desordenarse, o incluso duplicarse, e IP no manejará estas situaciones. El proporcionar estos servicios depende de prots superiores.

IP asume pocas cosas de las capas inferiores, sólo que los datagramas "probablemente" serán transportados al host de destino.

2.3.1 El datagrama IP

El datagrama IP es la unidad de transferencia en la pila IP. Tiene una cabecera con información para IP, y los datos relevantes para los prots superiores.

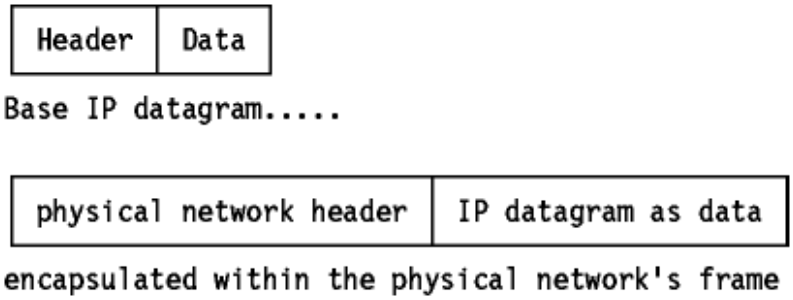


Figura: El datagrama IP

El datagrama IP está encapsulado en la trama de redd subyacente, que suele tener una longitud máxima, dependiendo del hardware usado. Para Ethernet, será típicamente de 1500 bytes. En vez de limitar el datagrama a un tamaño máximo, IP puede tratar la *fragmentación* y el *re-ensamblado* de sus datagramas. En particular, el IP est no impone una tamaño máximo, pero establece que todas las redes deberían ser capaces de manejar al menos 576 bytes. Los fragmentos de datagramas tienen todos una cabecera, copiada básicamente del datagrama original, y de los datos que la siguen. Se tratan como datagramas normales mientras son transportados a su destino. Nótese, sin embargo, que si uno de los fragmentos se pierde, todo el datagrama se considerará perdido, y los restantes fragmentos se considerarán perdidos.

2.3.1.1 Formato del datagrama IP

La cabecera del datagrama IP es de un mínimo de 20 bytes de longitud:

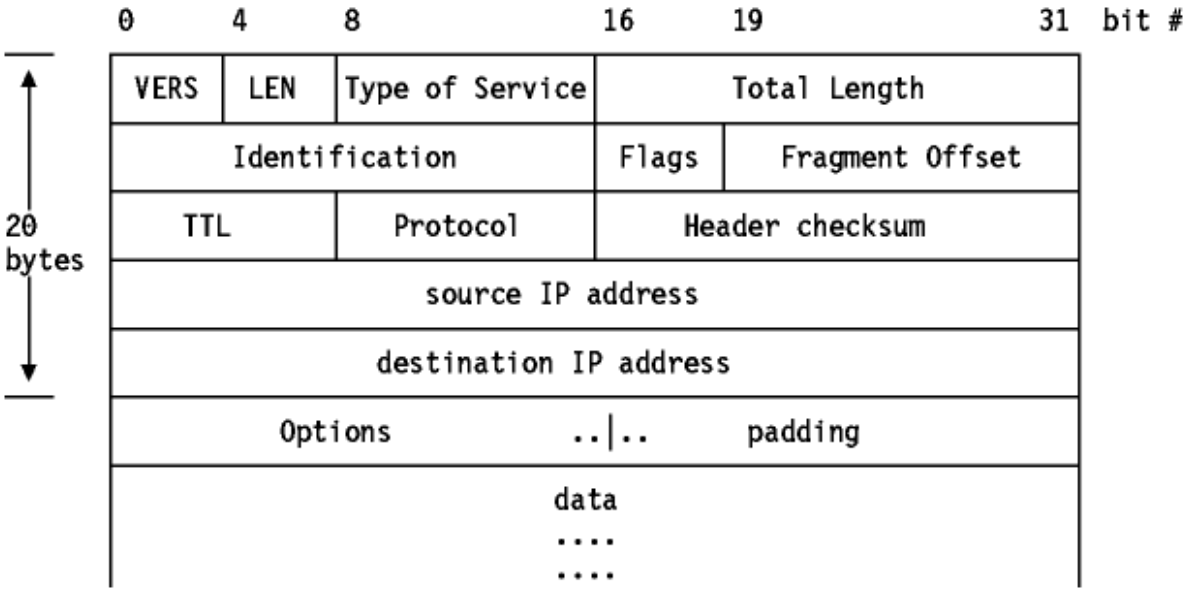


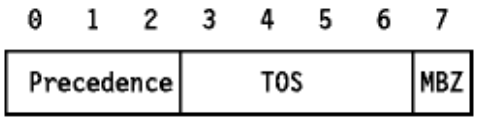
Figura: Formato del datagrama IP

Donde:

VERS
: La versión del protocolo IP. La versión actual es la 4. La 5 es experimental y la 6 es IPng(ver IP: La próxima generación(IPng)).

LEN
: La longitud de la cabecera IP contada en cantidades de 32 bits. Esto no incluye el campo de datos.

Type of Service
: El tipo de servicio es una indicación de la calidad del servicio solicitado para este datagrama IP.



Donde:

Precedencia
: Es una medida de la naturaleza y prioridad de este datagrama:

000
: Rutina

001
: Prioridad

010
: Imediato

011
: "Flash"

100
: "Flash override"

101
: Crítico

110
: Control de red("Internetwork control")

111
: Control de red("Network control")

TOS
: TOS("*type of service*"):

1000
: Minimizar retardo

0100
: Maximizar la densidad de flujo

0010
: Maximizar la fiabilidad

0001
: Minimizar el coste monetario

0000
: Servicio normal

MBZ
: Reservado para uso futuro(debe ser cero, a menos que participe en un experimento con IP que haga uso de este bit)
Una descripción detallada del TOS se puede encontrar en el RFC 1349.

Total Length
: La longitud total del datagrama, cabecera y datos, especificada en bytes.

Identification
: Un número único que asigna el emisor para ayudar a reensamblar un datagrama fragmentado. Los fragmentos de un datagrama tendrán el mismo número de identificación.

Flags
: Varios flags de control:

Donde:

0
: Reservado, debe ser cero

DF
: No fragmentar("Don't Fragment"): con 0 se permite la fragmentación, con 1 no.

MF
: Más fragmentos("More fragments"): 0 significa que se trata del último fragmento del datagrama, 1 que no es el último.

Fragment Offset
: Usado con datagramas fragmentados, para ayudar al reensamblado de todo el datagrama. El valor es el número de partes de 64 bits(no se cuentan los bytes de la cabecera) contenidas en fragmentos anteriores. En el primer(o único) fragmento el valor es siempre cero.

Time to Live
: Especifica el tiempo(en segundos) que se le permite viajar a este datagrama. Cada "router" por el que pase este datagrama ha de sustraer de este campo el tiempo tardado en procesarlo. En la realidad un "router" es capaz de procesar un datagrama en menos de 1 segundo; por ello restará uno de este campo y el TTL se convierte más en una cuenta de saltos que en una métrica del tiempo. Cuando el valor alcanza cero, se asume que este datagrama ha estado viajando en un bucle y se desecha. El valor inicial lo debería fijar el protocolo de alto nivel que crea el datagrama.

Protocol Number spotipprotn
: Indica el protocolo de alto nivel al que IP debería entregar los datos del datagrama. Algunos valores importantes son:

0
: Reservado

1
: ICMP("Internet Control Message Protocol")

2
: IGMP ("Internet Group Management Protocol")

3
: GGP("Gateway-to-Gateway Protocol")

4
: IP (IP encapsulation)

5
: Flujo("Stream")

6
: TCP("Transmission Control")

8
: EGP("Exterior Gateway Protocol")

9
: PIRP("Private Interior Routing Protocol")

17
: UDP ("User Datagram")

89
: OSPF("Open Shortest Path First")

La lista completa se puede encontrar en el *STD 2 - Números asignados de Internet*.

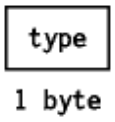
Header Checksum
: Es el checksum de la cabecera. Se calcula como el complemento a uno de la suma de los complementos a uno de todas las palabras de 16 bits de la cabecera. Con el fin de este cálculo, el campo checksum se supone cero. Si el checksum dde la cabecera no se corresponde con los contenidos, el datagrama se desecha, ya que al menos un bit de la cabecera está corrupto, y el ddatagrama podría haber llegado al destino equivocado.

Source IP Address
: La dirección IP de 32 bits del host emisor.

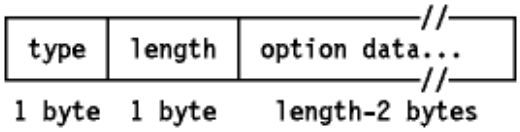
Destination IP Address
: La dirección IP de 32 bits del host receptor.

Options
: Longitud variable. No requiere que toda implementación de IP sea capaz de generar opciones en los datagramas que crea, pero sí que sea capaz de procesar datagramas que contengan opciones. El campo "Options" (opciones) tiene longitud variable. Puede haber cero o más opciones. Hay dos formatos para estas. El formato usado depende del valor del número de opción hallado en el primer byte.

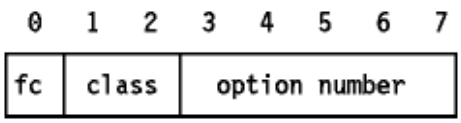
- Un byte de tipo("type byte") sólo.



- Un byte de tipo, un byte de longitud y uno o más bytes de opciones.



El byte de tipo tiene la misma estructura en ambos casos:



Donde:

fc
: "Flag copy", que indica si el campo de opcions se ha de copiar(1) o no(0) cuando el datagrama está fragmentado.

class

Un entero sin signo de 2 bits:

0
control

1
reservado

2
depurado y mediciones

3
reservado

option number
Entero sin signo de 5 bits.

0
Fin de la lista de opciones, con "class" a cero, fc a cero, y sin byte de longitud o de datos. Es decir, la lista termina con el byte X'00'. Sólo se requiere si la longitud de la cabecera IP(que es un múltiplo de 4 bytes) no se corresponde con la longitud real de las opciones.

1
No operación. Tiene "class" a cero, fc a cero y no hay byte de longitud ni de ddatagramaos. Es decir, un byte X'01' es NOP("no operation"). Se puede usar para alinear campos en el datagrama.

2
Securidad. Tiene "class" a cero, fc a uno y el byte de longitud a 11 y el de datos a 8. Se usa para la información de seguridad que necesitan las especificaciones del departamento de defensa de los US.

3
LSR("Loose Source Routing"). Tiene "class" a cero, fc a uno y hay un campo de datos de longitud variable.

4
IT("Internet Timestamp"). Tiene "class" a 2, fc a cero y hay un campo de datos de longitud variable.

7
RR("Record Route"). Tiene "class" a 0, fc a cero y hay un campo de datos de longitud variable.

8
SID("Stream ID", o identificador de flujo). Tiene "class" a 0, fc a uno y hay un byte de longitud a 4 y un byte de datos. Se usa con el sistema SATNET.

9
SSS("Strict Source Routing"). Tiene "class" a 0, fc a uno y hay un campo de datos de longitud variable.

length
cuenta la longitud(en bytes) de la opción, incluyendo los campo de tipo y longitud.

option data
no contiene datos relevantes para la opción.

padding
Si se usa una opción, el datagrama se rellena con bytes a cero hasta la siguiente palabra de 32 bits.

data
Los datos contenidos en el datagrama se pasan a un protocolo de nivel superior, como se especifica en el campo *protocol*.

2.3.1.2 Fragmentación

Cuando un datagrama IP viaja de un host a otro puede cruzar distintas redes físicas. Las redes físicas imponen un tamaño máximo de trama, llamado MTU(*"Maximum Transmission Unit")*, que limita la longitud de un datagrama. Por ello, existe un mecanismo para fragmentar los datagramas IP grandes en otros más pequeños, y luego reensamblarlos en el host de destino. IP requere que cada enlace tenga un MTU de al menos 68 bytes, de forma que si cualquier red proporciona un valor inferior, la fragmentación y el rensamblado tendrán que implementarse en la capa de la interfaz de red de forma transparente a IP. 68 es la suma de la mayor cabecera IP, de 60 bytes, y del tamaño mínimo posible de los datos en un fragmento(8 bytes). Las implementaciones de IP no están obligadas a manejar datagrama sin fragmentar mayores de 576 bytes, pero la mayoría podrá manipular valores más grandes, típicamente ligeramente más de 8192 bytes, o incluso mayores, y raramente menos de 1500.

Un datagrama sin fragmentar tiene a cero toda la información de fragmentación. Es decir, el flag fc y el fo(fragment offset) están a cero. Cuando se ha de realizar la fragmentación, se ejecutan los siguientes pasos:

- Se chequea el bit de flag DF para ver si se permite fragmentación. Si está a uno, el datagrama se desecha y se devuelve un error al emisor usando ICMP.
- Basándose en el valor MTU, el campo de datos se divide en dos o más partes. Todas las nuevas porciones de datos, excepto la última, se alinean a 8 bytes.
- Todas las porciones de datos se colocan en datagramas IP. Las cabeceras se copian de la cabecera original, con algunas modificaciones:
  - El bit de flag mf(more fragments) se pone a uno en todos los fragmentos, excepto en el último.
  - El campo fo se pone al valor de la localización de la porción de datos correspondiente en el at original, con respecto al comienzo del mismo. Su valor se mide en unidades de 8 bytes.
  - Si se incluyeron opciones en el datagrama original, el bit de orden superior del byte "type option" determina si se copiaran o no en todos los fragmentos o sólo en el primero. Por ejemplo, las opciones e encaminamiento de la fuente se tendrán que copiar en todos los fragmentos y por tanto tendrán a uno este bit.
  - Se inicializa el campo de longitud(length) del nuevo datagrama.
  - Se inicializa el campo de longitud(length) total del nuevo datagrama.
  - Se recalcula el checksum de la cabecera.
- Cada uno de estos datagramas se envía como un datagrama IP normal. IP maneja cada fragmento dde forma independiente, es decir, los fragmento pueden atravesar diversas rutas hacia su destino, y pueden estar sujetos a nuevas fragmentaciones si pasan por redes con MTUs inferiores.

En el host de destino, los datos se tienen que reensamblar. El host emisor inicializó el campo ID a un número único(dentro de los límites impuestos por el uso de un número de 16 bits). Como la fragmentación no altera este campo, los fragmentos que le van llegando al destino se pueden identificar, si este ID se usa junto con las direcciones IP fuente y destino(source, destination) del datagrama. También se chequea el campo de protocolo

Con el fin de reensamblar los fragmentos, el receptor destina un buffer de almacenamiento en cuanto llega el primer fragmento. Se inicia una rutina para un contador. Cuando el contador a un timeout y no se han recibido todos los datagramas, se desecha el datagrama. El valor inicial el contador es el TTL(time-to-live). Depende de la implementación, y algunas permiten configurarlo.

Cuando llegan los fragmentos siguientes, antes de que expire el tiempo, los datagramas se copian al buffer en la localización indicada por el fo(fragment offset). Cuando han llegado todos los datagramas, se restaura el datagrama original y continúa su procesamiento.

Nota: IP no proporciona el contador de reensamblado. Tratará cada datagrama, fragmentado o no, de la misma forma. Depende de una capa superior el implementar un timeout y reconocer la pérdida de fragmentos. Esta capa podría ser TCP para el transporte en un red orientada a conexión o UDP, para el caso contrario.

2.3.1.3 Opciones de encaminamiento del datagrama IP

El campo "options" del datagrama IP admite dos métodos para que el generador del datagrama dé explícitamente información de encaminamiento y uno para que el datagrama

determine a ruta que va a emplear.

LSR("Loose Source Routing")

Esta opción, conocida también como LSRR("Loose Source and Record Route"), proporciona un medio para que la fuente del dar suministre información de encaminamiento explícita que usarán los "routers" que retransmitan el datagrama, y para grabar la ruta seguida.

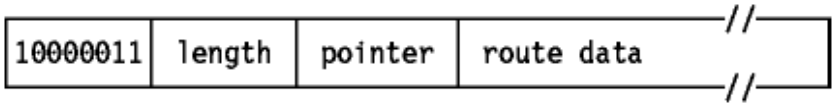


Figura: Opción LSR

1000011
: (131 decimal) es el valor del byte "option" para LSR.

length
: contiene la longitud de este campo, incluyendo los campos "type" y "length".

pointer
: apunta a los datagramas de la opción en la siguiente dirección IP a procesar. Es relativo al comienzo de la opción, por lo que su valor mínimo es de cuatro. Si su valor supera la longitud de la opción, se alcanza el final de la ruta de la fuente y el resto del encaminamiento se ha de basar en la dirección IP de destino(como en los datagramas que no tienen esta opción).

route data
: es una serie de direcciones IP de 32 bits.

Siempre que un datagrama llega a su destino y la ruta de la fuente no está vacía(pointer < length) el receptor:

- Tomará la siguiente dirección IP de este campo(el indicado por "pointer" y lo pondrá en el campo de la dirección IP de destino el datagrama.
- Pondrá la dirección IP local en la SL(source list) en la localización a la que apunte "pointer". La dirección IP local es la correspondiente a la red por la que se enviará el datagrama.
- Incrementará "pointer" en 4.
- Transmitirá el datagrama a la nueva dirección IP de destino.

Este procedimiento asegura que la ruta de retorno se graba en "route datagram"(en orden inverso) de modo que el receptor use estos datagramas para construir un LSR en el sentido inverso. Se denomina LSR(*"loose source route"*) porque al "router" retransmisor se le permite usar cualquier ruta y cualquier número de host intermedios para alcanzar la siguiente dirección de la ruta.

Nota: El host emisor pone la dirección IP del primer "router" intermedio en el campo dirección IP de destino y las direcciones de los demás "routers" de la ruta, incluyendo el destino, en la opción "source route". La ruta que hay grabada en el datagrama cuando este llega al objetivo contiene las direcciones IP de cada uno de los "routers" que retransmitió el datagrama.

SSR("Strict Source Routing")

Esta opción, llamada también SSRR(Strict Source and Record Route"), emplea el mismo principio que LSR exceptuando que el "router" intermedio *debe* enviar el datagrama a la siguiente dirección IP en la ruta especificada por la fuente a través de una red conectada directamente y no por medio de un "router" intermedio. Si no puede hacerlo, envía un mensaje ICMP "Destination Unreachable".

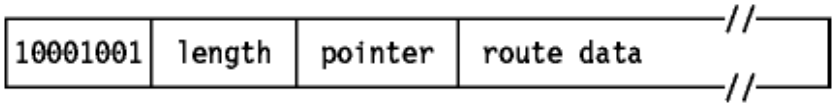


Figura: Opción SSR

1001001
: (137 decimal) es el valor del byte "option" para el método SSR

length
: tiene el mismo significado que para LSR

pointer
: tiene el mismo significado que para LSR

route data
: es una serie de direcciones IP

RR(Record Route)

Esta opción proporciona un medio para grabar la ruta de un datagrama IP. Funciona de modo similar al SSR anterior, pero en este caso el host fuente deja el campo de datos de encaminamiento vacío, que se irá llenando a medida que el datagrama viaja. Nótese que el host fuente debe dejar suficiente espacio para esta información: si el campo se llena antes de que el datagrama llegue a su destino, el datagrama se retransmitirá, pero se dejará de grabar la ruta.

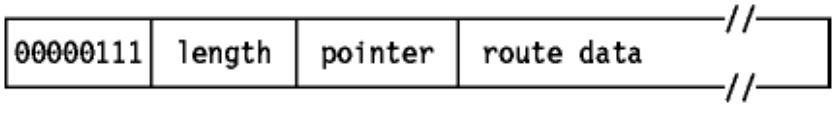


Figura: Opción RR

0000111
: (7 decimal) es el valor del byte "option" para el método RR

length
: tiene el mismo significado que para LSR

pointer
: tiene el mismo significado que para LSR

route data
: su longitud es un múltiplo de cuatro bytes, y lo elige el generador del datagrama

2.3.1.4 IT(Internet Timestamp)

El "timestamp " o sello de tiempo es una opción para forzar a algunos(o a todos) de los "routers" de la ruta hacia el destino a poner un "timestamp" en los datos de la opción. Los "timestamps" se miden en segundos y se pueden usar para la depuración. No se pueden emplear para medir el rendimiento por dos razones:

- No son lo bastante precisos porque la mayoría de los datagramas se envían en menos de un segundo.
- No son lo bastante precisos porque los "router" no han de tener relojes sincronizados.

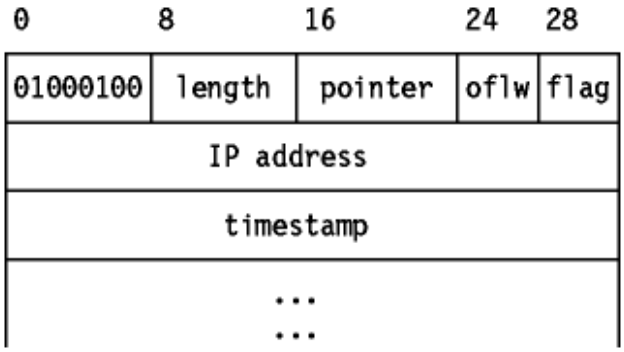


Figura: Opción IT

Donde

01000100
: (68 decimal) es el valor del byte "option" para IT.

length
: Contiene la longitud total de esta opción, incluyendo los campos "type" y "length".

pointer
: Apunta al siguiente "timestamp" a procesar(el primero que esté disponible).

oflw (overflow)
: Es un entero sin signo de 4 bits que indica el número de módulos IP que no pueden registrar "timestamps" por falta de espacio en el campo de datos.

flag
: Es una valor de 4 bits que indica cómo se han de registrar los "timestamps". Los valores posibles son:

0
: Sólo "timestamps", almacenados en palabras consecutivas de 32 bits.

1
: Cada "timestamp" se precede con la dirección IP del módulo que efectúa el registro.

2
: La dirección IP se pre-especifica, y un módulo IP sólo realiza el registro cuando encuentra su propia dirección en la lista.

timestamp
: Un "timestamp" de 32 bits medido en milisegundos desde la medianoche según UT (GMT).

El host emisor debe componer esta opción con un área de datos los bastante grande para almacenar todos los "timestamps". Si el área de los "timestamps" se llena, no se añaden más.

2.3.2 Encaminamiento IP

Una función importante de la capa IP es el *encaminamiento.* Proporciona los mecanismos básicos para interconectar distintas redes físicas. Esto significa que un host puede actuar simultáneamente como host normal y como "router".

Un "router" básico de este tipo se conoce como *"router" con información parcial de encaminamiento*, ya que sólo contiene información acerca de cuatro tipos de destino:

- Los hosts conectados directamente a una de las redes físicas a las que está conectado el "router"
- Los host o redes para las se le han dado al "router" definiciones específicas
- Los hosts o redes para las que el host ha recibido un mensaje ICMP *redirect*
- Un destino por defecto para todo lo demás

Los dos últimos casos permiten a un "router" básico comenzar con una cantidad muy limitada de información para irla aumentando debido a que un "router" más avanzado lance un mensaje ICMP *redirect* cuando reciba un datagrama y conozca un "router" mejor en la misma red al que dirigir el datagrama. Este proceso se repite cada vez que un "router" básico se reinicia.

Se necesitan protocolo adicionales para implementar un "router" completamente funcional que pueda intercambiar información con otros "routers" en redes remotas. Tales "routers" son esenciales, excepto en redes pequeñas, y los protocolos que usan se discuten en Protocolos de encaminamiento.

2.3.2.1 Destinos directos e indirectos

Si el host de destino está conectado a una red a la que también está conectado el host fuente, un datagrama IP puede ser enviado directamente, simplemente encapsulando el datagrama IP en un trama. Es lo que se llama *encaminamiento directo*.

El *encaminamiento indirecto* ocurre cuando el host de destino no está en una red conectada directamente al host fuente. La única forma de alcanzar el destino es a través de uno o más "routers". La dirección del primero de ellos(el *primer salto*) se llama *ruta indirecta*. La dirección del primer salto es la única información que necesita el host fuente: el "router" que reciba el datagrama se responsabiliza del segundo salto, y así sucesivamente.

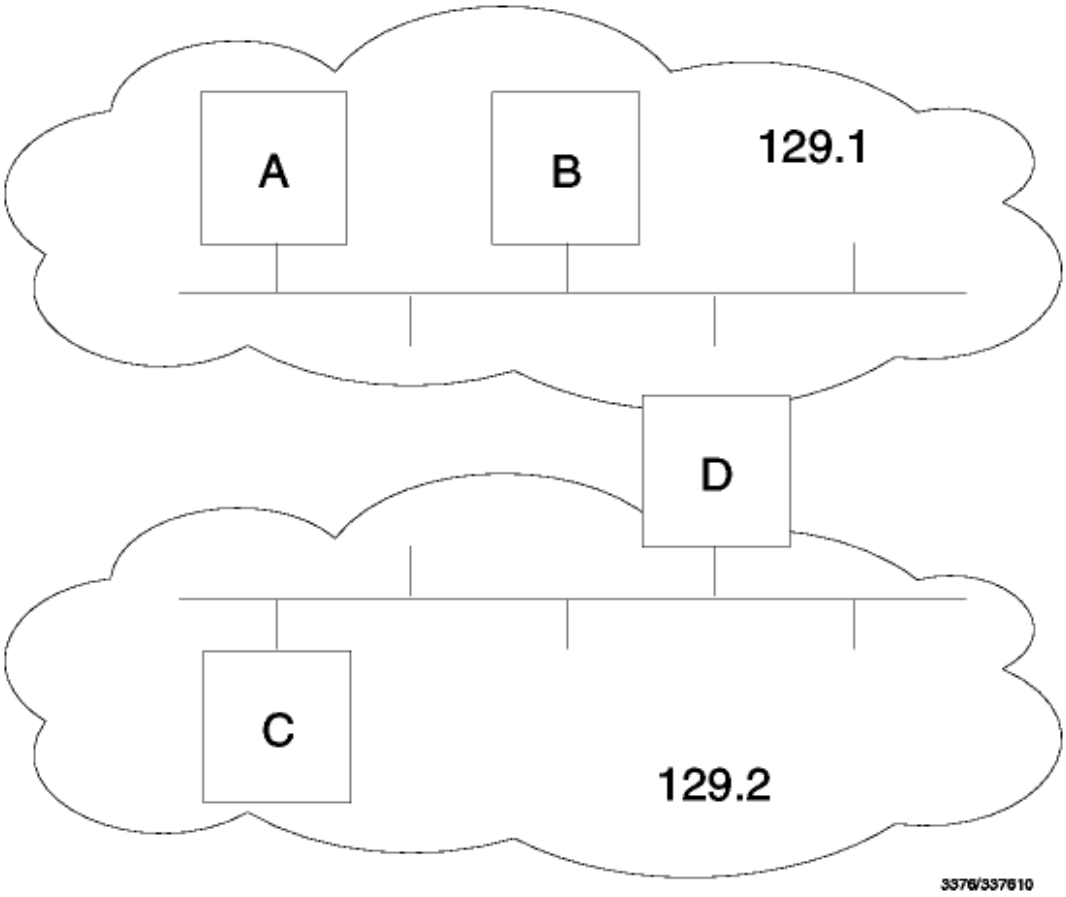


Figura: Rutas IP directas e indirectas - El host A tiene una ruta directa con B y D, y una indirecta con C. El host D es un "router" entre las redes 129.1 y 129.2.

Un host puede distinguir si una ruta es directa o indirecta examinando el número de red y de subred de la dirección IP.

1. Si coinciden con una de las direcciones IP del host fuente, la ruta es directa.

   El host necesita ser capaz de direccionar correctamente el objetivo usando un protocolo inferior a IP. Esto se puede hacer automáticamente, usando un protocolo como ARP(ver ARP("Address Resolution Protocol")), que se usan en LANs con broadcast, o estáticamente y configurando el host, por ejemplo cuando un host MVS tiene una conexión TCP/IP sobre un enlace SNA.

2. Para rutas indirectas, el único conocimiento requerido es la dirección IP de un "router" que conduzca a la red de destino.

Las implementaciones de IP pueden soportar también rutas explícitas, es decir, una ruta a una dirección IP concreta. Esto es habitual en las conexiones que usan SLIP("Serial Line Internet Protocol") que no proporciona un mecanismo para que dos hosts se informen mutuamente de sus direcciones IP. Tales rutas pueden tener incluso el mismo número de red que el host, por ejemplo en subredes compuestas de enlaces punto a punto. En general, sin embargo, la información de encaminamiento se genera sólo mediante los números de red y de subred.

2.3.2.2 Tabla de encaminamiento IP

Cada host guarda el conjunto de mapeados entre las direcciones IP de destino y las direcciones IP del siguiente salto para ese destino en una tabla llamada *tabla de encaminamiento IP*.

En esta tabla se pueden encontrar tres tipos de mapeado:

1. Rutas directas, para redes conectadas localmente
2. Rutas indirectas, para redes accesibles a través de uno o más "routers"
3. Un ruta por defecto, que contiene la ir IP de un "router" que todas las direcciones IP no contempladas en las rutas directas e indirectas han de usar.

Ver la red en Figura - Ejemplo de tabla de encaminamiento IP para un ejemplo.

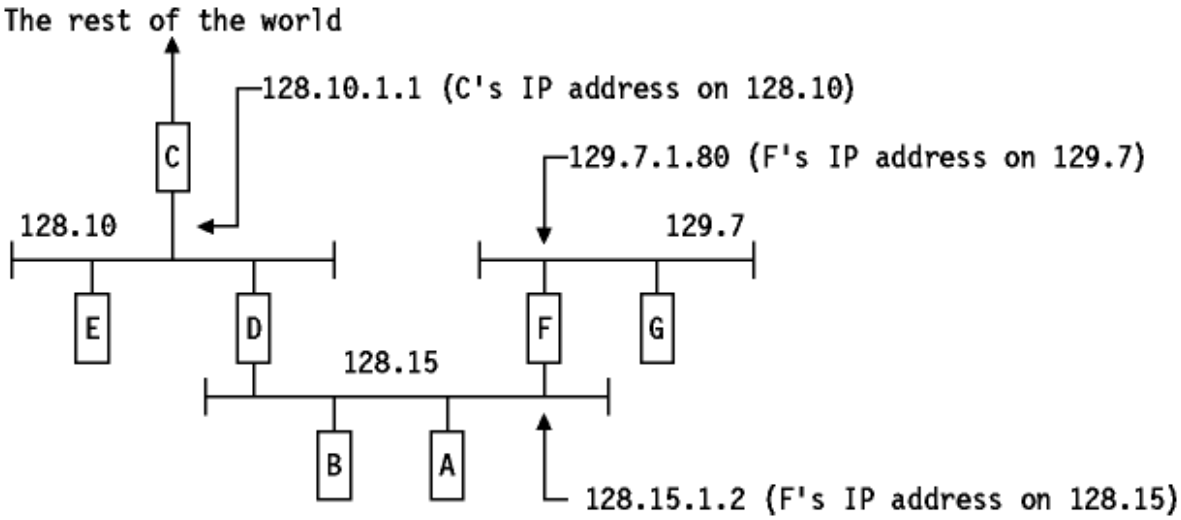


Figura: Ejemplo de tabla de encaminamiento IP

La tabla de encaminamiento contiene las siguientes entradas

Destination

---

route via
128.10
direct attachment
128.15
direct attachment
129.7
128.15.1.2
default
128.10.1.1

2.3.2.3 Algoritmo de encaminamiento IP

De los principios ya comentados de IP, es fácil deducir los pasos que IP debe tomar con el fin de determinar la ruta para un datagrama de salida. Es lo que se denomina *algoritmo de encaminamiento IP*, y se muestra esquemáticamente en Figura - Algoritmo de encaminamiento IP.

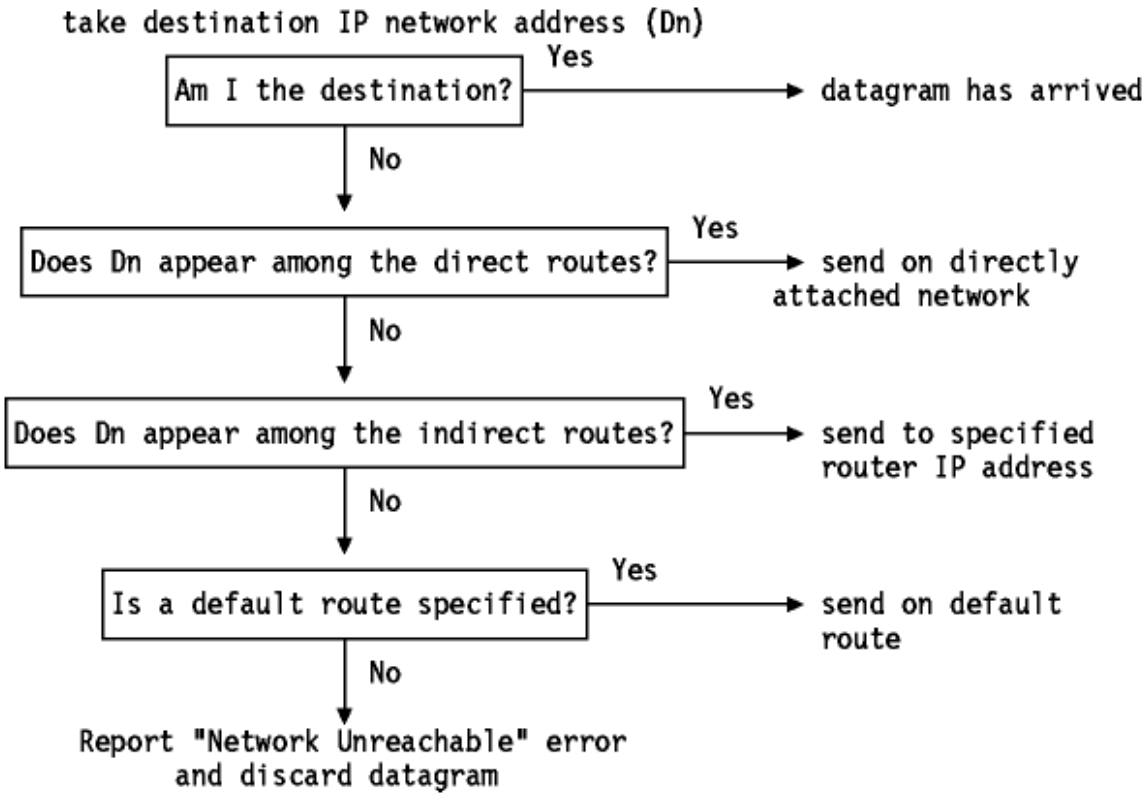


Figura: Algoritmo de encaminamiento IP

Nótese que se trata de un proceso iterativo. Se aplica a todo host que maneje un datagrama, exceptuando al host al que se entrega finalmente el datagrama.

Tabla de contenidos ICMP("Internet Control Message Protocol")

2.4 ICMP("Internet Control Message Protocol")

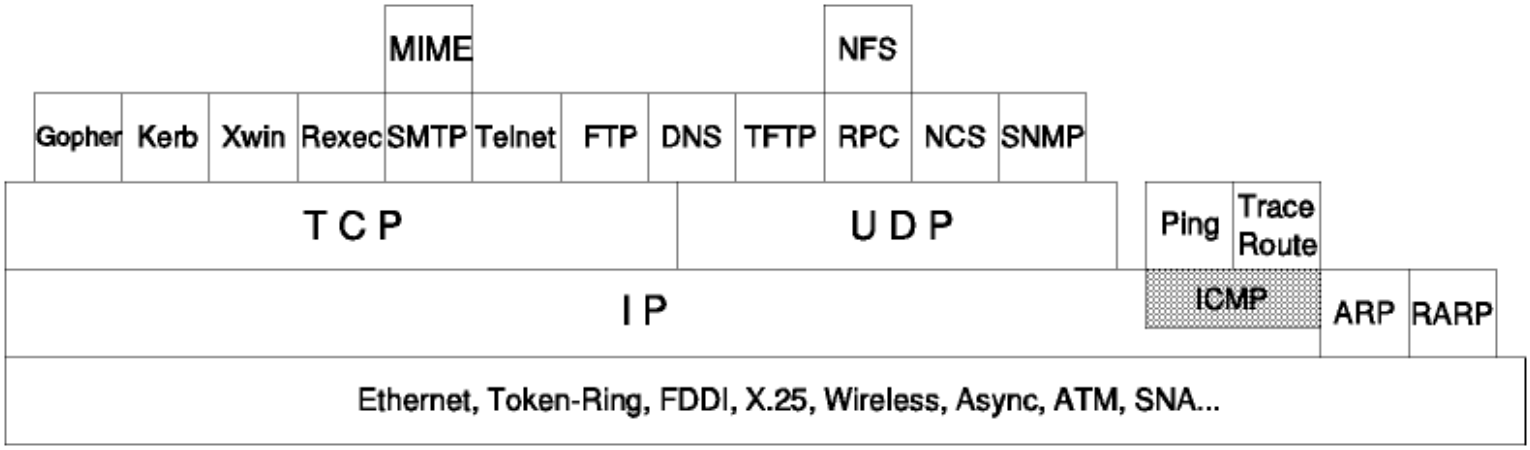


Figura: ICMP("Internet Control Message Protocol")

ICMP es un *protocolo estándar* con número STD 5, que además incluye IP(ver IP("Internet Protocol")) e IGMP (ver IGMP("Internet Group Management")). Su status es *requerido*. Se describe en el RFC 792, actualizado en el RFC 950.

"Path MTU Discovery" es un *protocolo estándar provisional* con status *electivo*. Se describe en el RFC 1191.

ICMP "Router Discovery" es *un protocolo propuesto como estándar* con status *electivo*. Es descrito en el RFC 1256.

Cuando un "router" o un host de destino debe informar al host fuente acerca del procesamiento de datagramas, utiliza el ICMP("Internet Control Message Protocol"). ICMP puede caracterizarse del modo siguiente:

- ICMP usa IP como si ICMP fuera un protocolo del nivel superior(es decir, los mensajes ICMP se encapsulan en datagramas IP). Sin embargo, ICMP es parte integral de IP y debe ser implementado por todo módulo IP.
- ICMP se usa para informar de algunos errores, *no* para hacer IP fiable. Aún puede ocurrir que los datagramas no se entreguen y que no se informe de su pérdida. La fiabilidad debe ser implementada por los protocolos de nivel superior que usan IP.
- ICMP puede informar de errores en cualquier datagrama IP con la excepción de mensajes IP, para evitar repeticiones infinitas.
- Para datagramas IP fragmentados, los mensajes ICMP sólo se envían para errores ocurridos en el fragmento cero. Es decir, los mensajes ICMP nunca se refieren a un datagrama IP con un campo de desplazamiento de fragmento.
- Los mensajes ICMP nunca se envían en respuesta a datagramas con una dirección IP de destino que sea de broadcast o de multicast.
- Los mensajes ICMP nunca se envían en respuesta a un datagrama que no tenga una dirección IP de origen que represente a un único host. Es decir, la dirección de origen no puede ser cero, una dirección de looopback, de broadcast o de multicast.
- Los mensajes ICMP nunca se envían en respuesta a mensajes ICMP de error. Pueden enviarse en respuesta a mensajes ICMP de consulta(los tipos de mensaje ICMP 0, 8, 9, 10 y 13 al 18).
- El RFC 792 establece que los mensajes ICMP "pueden" ser generados para informar de errores producidos en el procesamiento de datagramas IP, *no* que "deban". En la práctica, los "routers" generarán casi siempre mensajes ICMP para los errores, pero en el caso de los host de destino, el número de mensajes ICMP generados es una cuestión de implementación.

2.4.1 Mensajes ICMP

Los mensajes ICMP se describen en los RFCs 792 y 950, correspondientes al STD 5 y son obligatorios.

Los mensajes ICMP se envían en datagramas IP. La cabecera IP siempre tendrá un número de protocolo de 1, indicando que se trata de ICMP y un servicio de tipo 0(rutina). El campo de datos de IP contendrá el auténtico mensaje ICMP en el formato mostrado en Figura - Formato de mensajes ICMP.

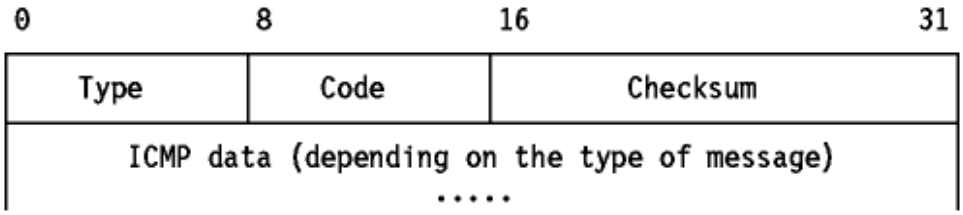


Figura: Formato de mensajes ICMP

Donde:

Type
: Especifica el tipo del mensaje:

0
: Echo reply

3
: Destination unreachable

4
: Source quench

5
: Redirect

8

Echo

9
Router Advertisement

10
Router Solicitation

11
Time exceeded

12
Parameter Problem

13
Timestamp request

14
Timestamp reply

15
Information request(obsolete)

16
Information reply(obsolete)

17
Adress mask request

18
Adress mask reply

Code
Contiene el código de error para el datagrama del que da parte el mensaje ICMP. La interpretación depende del tipo de mensaje.

Checksum
Contiene el complemento a 1 de 16 bits de la suma del "ICMP message starting with the ICMP Type field". Para computar este checksum se asume en principio que su valor es cero. Este algoritmo es el mismo que el usado por IP para el cálculo de la cabecera IP. Compárese con el algoritmo de UDP y TCP(ver UDP("User Datagram Protocol") y TCP("Transmission Control Protocol")) que incluyen además una *pseudocabecera-IP* en el checksum.

Data
Contiene información para el mensaje ICMP. Típicamente se tratará de parte del mensaje IP original para el que se generó el mensaje ICMP. La longitud de los datos puede calcularse como la diferencia entre la longitud del datagrama IP que contiene el mensaje y la cabecera IP.

Cada uno de los mensajes se explica abajo.

2.4.1.1 Echo Reply (0)

Ver Echo (8) y Echo Reply (0).

2.4.1.2 Destination Unreachable (3)

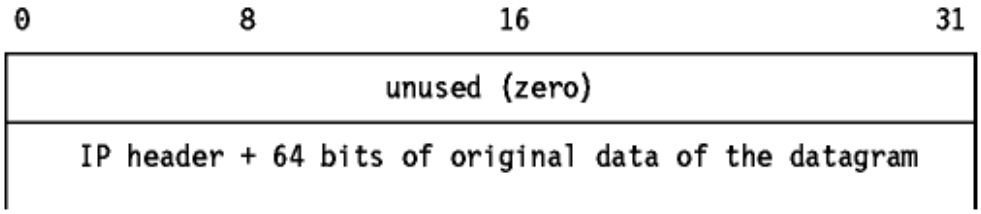


Figure: destination unreachable o " destino inalcanzable" de ICMP

Si este mensaje es recibido de un "router" intermediario, significa que el "router" considera la dirección IP de destino como inalcanzable.

Si se recibe este mensaje del host de destino, significa que el protocolo especificado en el campo de número de protocolo del datagrama original no está activo, que ese protocolo no está activo en ese host o bien que es el puerto indicado el que no está activo(ver UDP("User Datagram Protocol") para una introducción al concepto de puerto).

El campo de código de cabecera tendrá uno de los siguientes valores:

0
network unreachable

1
host unreachable

2
protocol unreachable

3
port unreachable

4
fragmentation needed but the *Do Not Fragment* bit was set

5
source route failed

6
destination network unknown

7
destination host unknown

8
source host isolated (obsolete)

9
destination network administratively prohibited

10
destination host administratively prohibited

11
network unreachable for this type of service

12
host unreachable for this type of service

13
communication administratively prohibited by filtering

14
host precedence violation
15
precedence cutoff in effect

Si un "router" implementa el protocolo de resolución de caminos MTU, el formato del mensaje "Destination unreachable" se cambia por el código 4 para incluir el MTU del enlace que no pudo aceptar el datagrama.

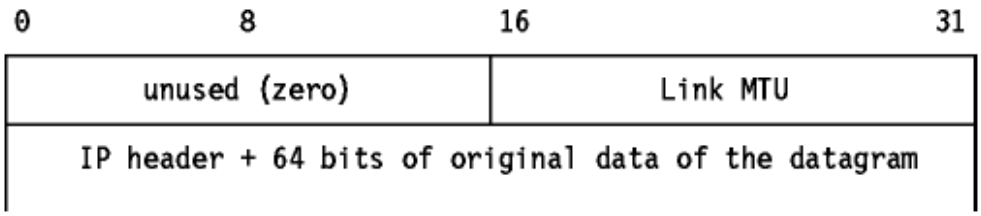


Figura: fragmentación de ICMP requerida con el enlace MTU

2.4.1.3 Source Quench (4)

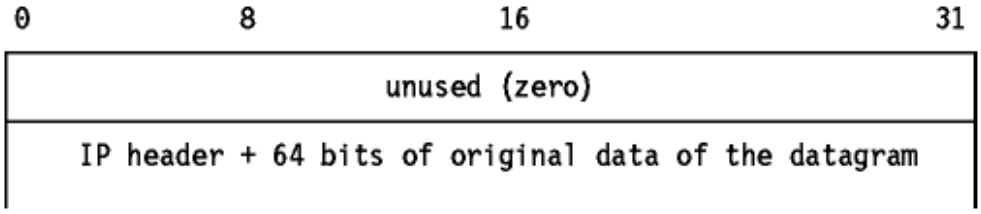


Figure: Source Quench de ICMP

Si se recibe este mensaje de un router intermedio, significa que el "router" no dispone de suficiente espacio en el buffer para encolar los datagramas de salida para la siguiente red.

Si este mensaje procede del host de destino, significa que los datagramas entrantes llegan demasiado rápidos para ser procesados.

El código de la cabecera ICMP siempre es cero.

2.4.1.4 Redirect (5)

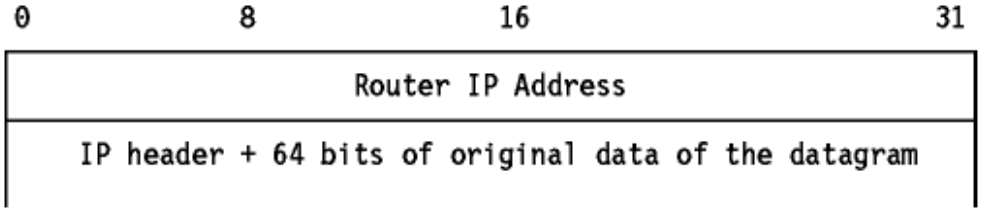


Figura: redirect de ICMP

Si se recibe este mensaje de un "router" intermedio, significa que el host debería los siguientes datagramas para esa red al "router" cuya dirección IP se especifica en el mensaje ICMP. Este otro "router" habrá de estar siempre en la misma subred que el host que envió el datagrama y el que lo devolvió. Enviará el datagrama a su siguiente dirección de salto; si la dirección del "router" coincide con la dirección fuente del datagrama original, indica un bucle. Este mensaje ICMP no se enviará si el datagrama IP contiene un ruta fuente.

La cabecera ICMP tendrá uno de los siguientes valores:

0
Network redirect
1
Host redirect
2
Network redirect for this type of service
3
Host redirect for this type of service

2.4.1.5 Echo (8) y Echo Reply (0)

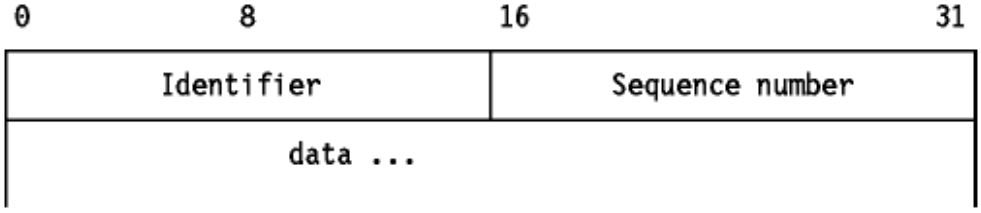


Figura: ICMP Echo y Echo Reply

Echo se usa para detectar si otro host está activo en la red. La fuente inicializa el identificador y el número de secuencia(que se utiliza cuando se envían múltiples mensajes "echo request"), añade algunos datos al campo de datos y envía el "echo" ICMP al host de destino. El código de la cabecera ICMP es cero. El receptor cambia el tipo del mensaje a "echo reply" y devuelve el datagrama al host fuente. El comando Ping emplea este mecanismo paca determinar si es posible alcanzar a un host de destino(ver Ping).

2.4.1.6 Router Advertisement (9) y Router Solicitation (10)

Los mensajes ICMP 9 y 10 son opcionales. Se describen en el RFC 1256, que es electivo.

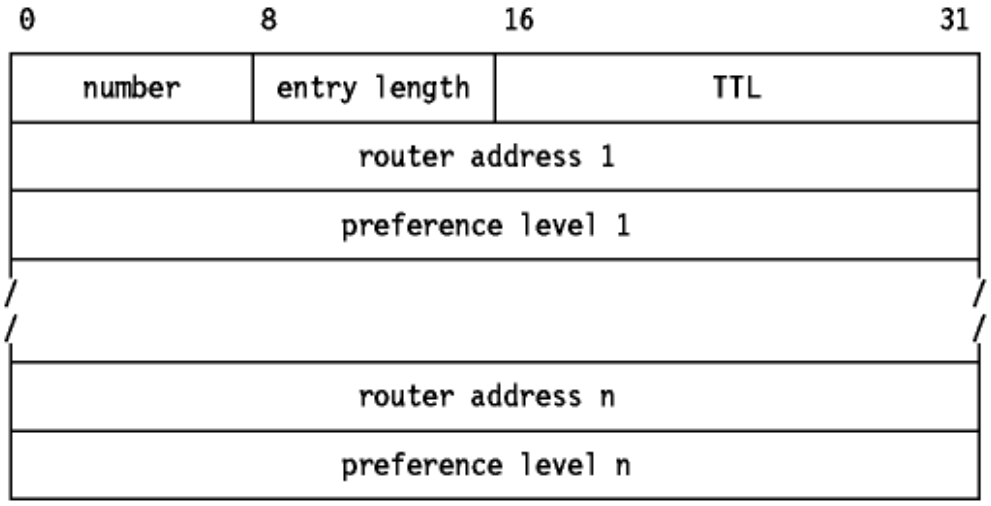


Figura: Router Advertisement de ICMP

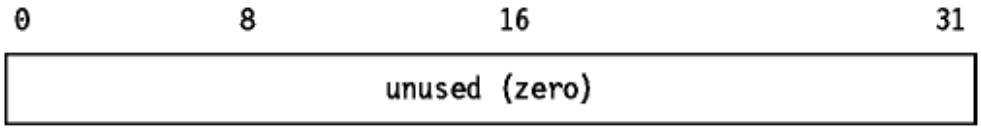


Figure: Router Solicitation de ICMP

number
: El número de entradas del mensaje.

entry length
: La longitud de una entrada en unidades de 32 bits. Vale 2 (32 bits para la dirección IP y 32 bits para el valor tomado por preferencia).

TTL
: El número de segundos que se considerará válida una entrada.

router address
: Una de las direcciones IP del host fuente.

preference level
: Un nivel expresado con un valor de 32 bits con signo que indica la preferencia a asignar a esta dirección al seleccionar un "router" por defecto para una subred. Cada "router" de una subred es responsable de anunciar su propio nivel de preferencia. La preferencia aumenta cuanto mayor es el valor, y viceversa. El valor por defecto es cero, que está en el centro del rango de valores. Un valor de X'80000000' -2exp31 indica que el "router" no se debería usar jamás como "router" por defecto.

La cabecera ICMP es cero para ambos mensajes.

Estos dos mensajes se usan si un host o un "router" soporta el RDP("Router Discovery Protocol"). El uso del multicast está recomendado, pero se puede usar el broadcast si la interfaz no soporta el multicast. Los "router" anuncian periódicamente sus direcciones IP en subredes si han sido configurados para que lo hagan. Los anuncios se hacen en la dirección de multicast(224.0.0.1) o de broadcast limitado(255.255.255.255). El comportamiento por defecto es enviar anuncios cada 10 minutos con un TL e 1800(30 minutos). Los "routers" también responden a los mensajes de solicitud que puedan recibir. Pueden responder directamente al solicitante, o esperar un intervalo de tiempo aleatorio y relativamente corto y responder con un multicast. Los hosts pueden enviar solicitudes hasta que reciben una respuesta. Las solicitudes se envían a la dirección de multicast para todos los "routers"(224.0.0.2) o a la de broadcast limitado(255.255.255.255). Típicamente, tres mensajes de solicitud se envían a intervalos de 3 segundos. Alternativamente, un host puede esperar a los anuncios efectuados periódicamente. Cada vez que un host recibe un anuncio, actualiza su "router" por defecto si el nuevo anuncio tiene una preferencia superior y fija el TTL para que la entrada se ajuste al valor del nivel de preferencia. Cuando el host recibe un nuevo valor para su "router" por defecto actual, pone el valor TTL al del nuevo anuncio. Esto proporciona además un mecanismo para que los "router" se declaren no disponibles: envían un anuncio con un TTL de cero.

2.4.1.7 Time Exceeded (11)

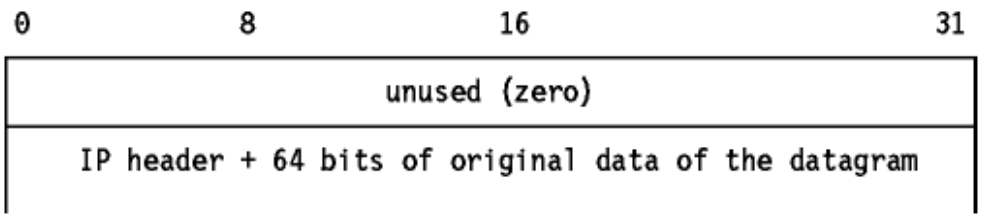


Figura: Time Exceeded de ICMP

Si se recibe este mensaje de un "router" intermedio, significa que el TTL de un datagrama IP ha expirado.

Si se recibe del host de destino, significa que el TTL para ensamblar el datagrama ha expirado mientras el host esperaba uno de sus fragmentos. La cabecera ICMP puede tener

uno de los siguientes valores:

0
: transit TTL exceeded

1
: reassembly TTL exceeded

2.4.1.8 Parameter Problem (12)

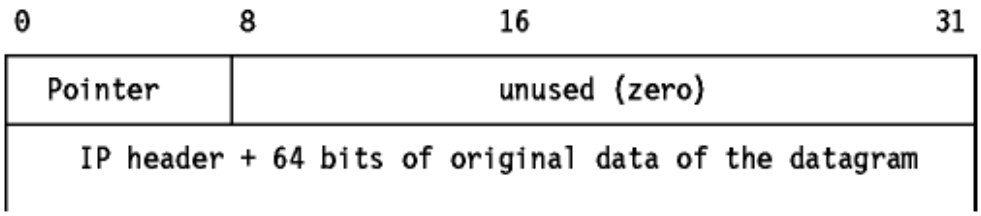


Figura: Parameter Problem de ICMP

Indica que se encontró un problema durante el procesamiento de los parámetros de la cabecera IP. El campo puntero apunta al byte del datagrama original en el que se encontró el problema. La cabecera ICMP puede tener uno de los siguientes valores:

0
: unspecified error

1
: required option missing

2.4.1.9 Timestamp Request (13) y Timestamp Reply (14)

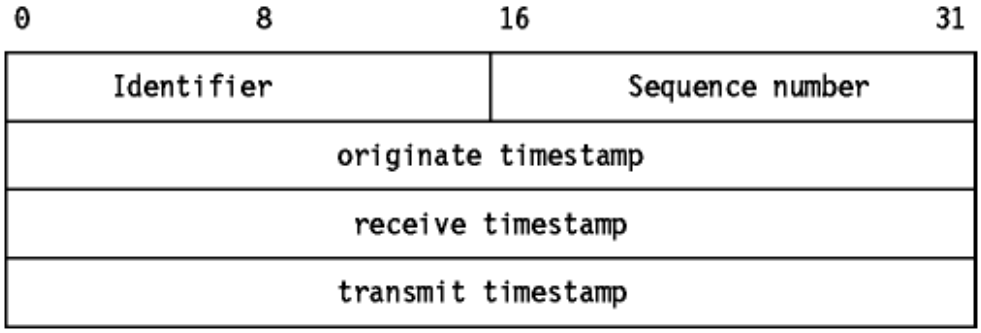


Figura: Timestamp Request y Timestamp Reply ICMP

Estos dos mensajes se emplean para medir el rendimiento y para la depuración. No se emplean para la sincronización: para eso está el NTP("Network Time Protocol")(ver Protocolos Time y Daytime).

El host fuente envía el identificador y el número de secuencia(usado si se envían múltiples mensajes "timestamp requests"), fija su sello de tiempo y se lo envía al receptor. El host receptor fija el valor de los sellos de tiempo de recepción y de envío, cambia el tipo del mensaje a "timestamp reply" y se lo devuelve al receptor. El receptor dispone de dos sellos de tiempo en caso de que haya una diferencia sensible entre los tiempos de recepción y de transmisión, aunque en la práctica la mayoría de las implementaciones efectuarán ambas operaciones(recepción y respuesta) de una sola vez, dando a los dos sellos el mismo valor. Los sellos de tiempo indican el número de milisegundos trasncurridos desde la medianoche según el meridiano de Greenwich(GMT).

2.4.1.10 Information Request (15) e Information Reply (16)

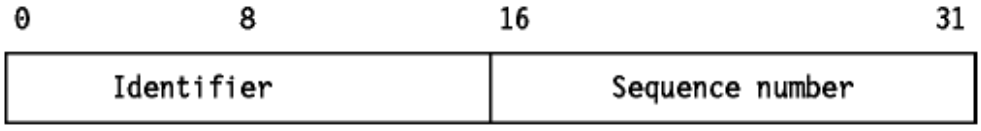


Figura: Information Request e Information Reply de ICMP

El mensaje Information Request lo lanza un host para obtener una dirección IP para una red con la que está conectado. El host fuente envía la solicitud con la dirección IP de destino puesta a cero en la cabecera IP(refiriéndose a su propia red) y espera una respuesta de un servidor autorizado a asignar direcciones IP a otros hosts. La cabecera ICMP vale cero. La respuesta contendrá la direcciones IP de red en los campos de dirección fuente y dirección de destino de la cabecera IP. Este mecanismo está obsoleto. Ver RARP("Reverse Address Resolution Protocol").

2.4.1.11 Address Mask Request (17) y Address Mask Reply (18)

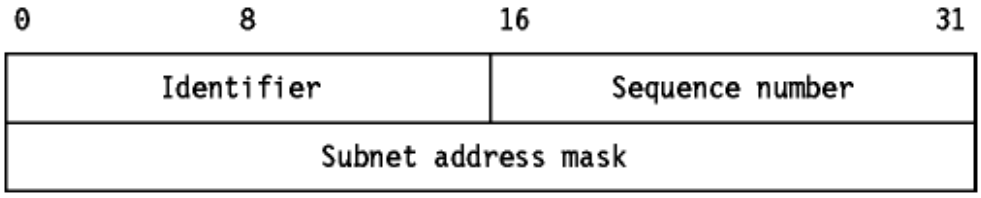


Figura: Address Mask Request y Reply de ICMP

El mensaje Address Mask Request es usado por un host cuando quiere determinar qué máscara de subred usar la red a la que está conectado. La mayoría de los hosts se configurarán con su máscara de subred, pero algunos, tales como una estación de trabajo sin disco, deben obtener esta información de un servidor. Un host utiliza RARP (ver RARP("Reverse Address Resolution Protocol")) para obtener su dirección IP. Para obtener una máscara de subred, el host hace un broadcast del mensaje Address Mask Request. Cualquier host en la red que se haya configurado para enviar mensajes Address Mask Reply rellenará esta máscara, convertirá el tipo del mensaje a Address Mask Reply y se lo devolverá al host fuente. La cabecera ICMP tiene valor cero.

2.4.2 Aplicaciones de ICMP

Hay dos aplicaciones simples y muy extendidas basadas en ICMP: el Ping y el Traceroute. El Ping usa los mensajes ICMP Echo y Echo Reply para determinar si un host es alcanzable. El Traceroute envía datagramas IP con bajos TTLs para que expiren durante la ruta que les dirige al destino. Utiliza los valores de los mensajes ICMP Time Exceeded para determinar en que parte de la red expiraron los datagramas y reconstruye así un esquema de la ruta hasta el host de destino. Estas aplicaciones se explican más detalladamente en Ping y Traceroute.

2.4.3 ICMP para la versión 6 de IP

La implementación de ICMP explicada arriba es específica de la versión 4 de IP(IPv4). La versión 6 de IP(IPv6, ver IPv6P) requerirá una nueva versión de ICMP. Las definiciones de ambas no están completas aún. Ya se conocen algunas características importantes:

- ICMP para IPv6 usará un nuevo número de protocolo para distinguirlo de ICMPv4.
- El formato de la cabecera ICMP permanecerá igual.
- Las longitudes de los campos de los mensajes cambiará para ajustarse a los mensajes IPv6, que serán de mayor longitud.
- Los valores "Type" y "Code" cambiarán. Algunos valores poco usados se eliminarán.
- El tamaño de los mensajes ICMP aumentará con el fin de explotar el tamaño máximo aumentado de los paquetes que IPv6 puede trasmitir sin fragmentar.
- La variante "Fragmentation Required" del mensaje ICMP "Destination unreachable" será reemplazado por el mensaje "Packet Too Big" que incluirá la MTU("Maximum Transmission Unit") de salida en la que se ha localizado el problema.
- IGMP (ver IGMP("Internet Group Management Protocol")) se fundirá con ICMP.

Tabla de contenidos Ping

2.5 Ping

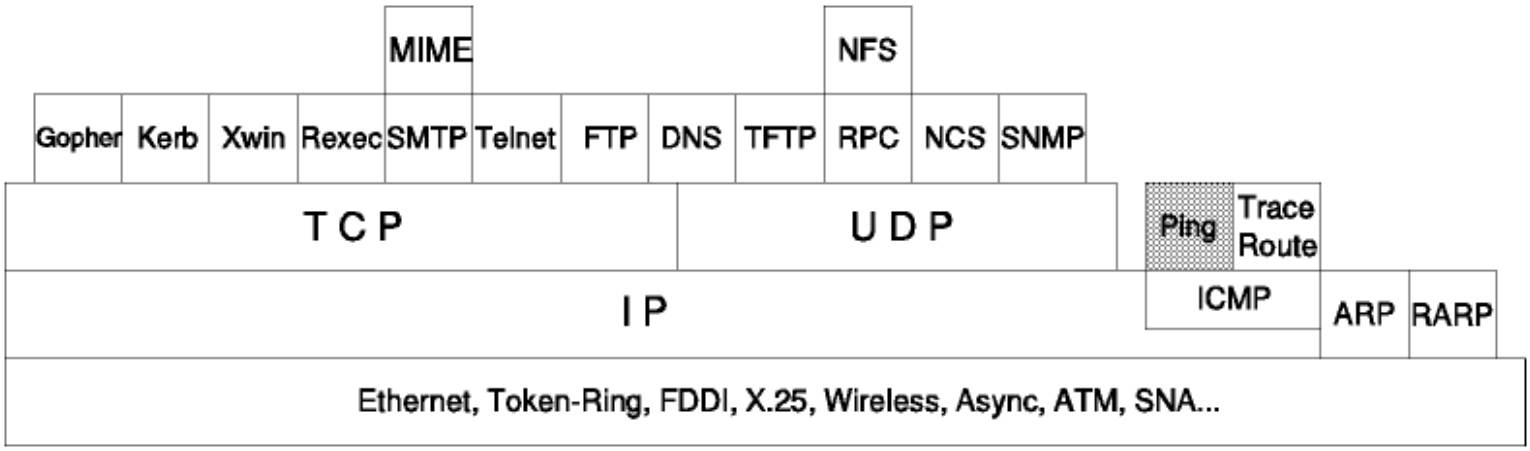


Figura: PING(Packet InterNet Groper)

El ping es la más sencilla de todas las aplicaciones TCP/IP. Envía uno o más datagramas a un host de destino determinado solicitando una respuesta y mide el tiempo que tarda en retornarla. La palabra *ping*, que puede usarse como nombre o como verbo(en inglés), procede de la operación de sónar empleada para localizar un objeto submarino. También es un acrónimo de *Packet InterNet Groper*.

Tradicionalmente, si podías hacer un ping a un host, otras aplicaciones como Telnet o FTP podían comunicarse con ese host. Con el advenimiento de las medidas de seguridad en Internet, particularmente los cortafuegos("firewalls"; ver Cortafuegos("Firewalls") para más información), que controlan el acceso a redes a través del protocolo de aplicación y/o el número de puerto, esto ha dejado de ser estrictamente cierto. Aún así, el primer test para comprobar si es posible alcanzar un host sigue siendo intentar hacerle un ping.

La sintaxis usada en diferentes implementaciones varía de unas plataformas a otras. La mostrada aquí corresponde a la implementación en OS/2:

```
ping [-Opción] host [Tamaño [Paquetes]]
```

Donde:

Opción
: Activa diversas opciones del ping.

host
: El destino: un nombre simbólico o bien una dirección IP.

Tamaño.
: El tamaño del paquete.

Paquetes.
: El número de paquetes a enviar.

Ping usa los mensajes Eco y Respuesta al Eco("Echo", "Echo Reply") de ICMP, como se describe en "Echo" (8) y "Echo Reply" (0). Ya que se requiere ICMP en cada implementación de TCP/IP, a los hosts no les hace falta un servidor separado para responder a los pings.

El ping es útil para verificar instalaciones TCP/IP. Consideremos las cuatro siguientes formas del comando; cada una requiere operar con una parte adicional de la instalación TCP/IP.

ping loopback
: Verifica la operatividad del software de la base de TCP/IP.

ping my-IP-address
: Verifica si el correspondiente dispositivo de la red física puede ser direccionado.

ping a-remote-IP-address
: Verifica si es posible acceder a la red.

ping a-remote-host-name
: Verifica la operatividad del servidor de nombres(o del "flat namespace resolver", dependiendo de la instalación) .

Tabla de contenidos Ping

2.6 "Traceroute"

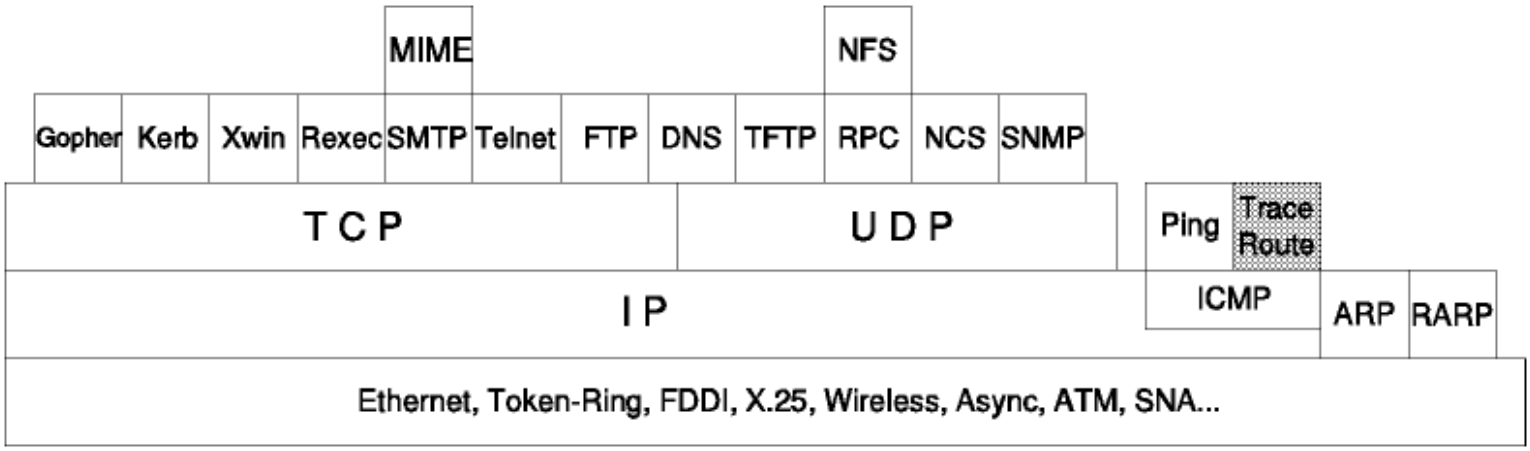


Figura: "Traceroute"

El programa "traceroute" puede ser útil cuando se usa para la depuración. Nos permite determinar la ruta que siguen los datagramas IP de host a host.

El "traceroute" se basa en ICMP. Envía un datagrama IP con un tiempo de vida(TTL) de 1 al host de destino. El primer "router" que vea el datagrama decrementará el TTL a 0 y devolverá el mensaje ICMP "Tiempo excedido"("Time Exceeded"), además de eliminar el datagrama. De este modo se identifica el primer "router" del camino. Este proceso se puede repetir sucesivamente con valores mayores del TTL con el fin de identificar la serie de "routers" que se encuentran en el camino hasta el host de destino. En realidad, el "traceroute" envía al host de destino datagramas UDP que referencian un número de puerto que está fuera del rango usado normalmente. Esto permite al "traceroute" determinar cuando se ha alcanzado el host de destino, es decir, cuando recibe el mensaje ICMP "Puerto inalcanzable"("Port Unreachable").

Tabla de contenidos IGMP(Internet Group Management Protocol)

2.7 IGMP("Internet Group Management Protocol")

IGMP es un *protocolo estándar* con un número STD de 5 que incluye además a IP(ver IP("Internet Protocol")) e ICMP (ver ICMP("Internet Control Message Protocol")). Su status es *recomendado* y el RFC 1112 lo describe.

Note: IP e ICMP son *requeridos*.

IGMP se considera más bien como una extensión de ICMP y ocupa el mismo lugar en la pila de protocolos IP. (3)

Ver Multicasting para una introducción al multicasting.

2.7.1 Mensajes IGMP

Los mensajes ICMP se envían en datagramas IP. La cabecera IP siempre tendrá un número de protocolo igual a 2, indicando IGMP y un tipo de servicio cero(rutina). El campo de datos IP contendrá el mensaje IGMP de ocho bytes mostrado en Figura - Formato del mensaje ICMP.

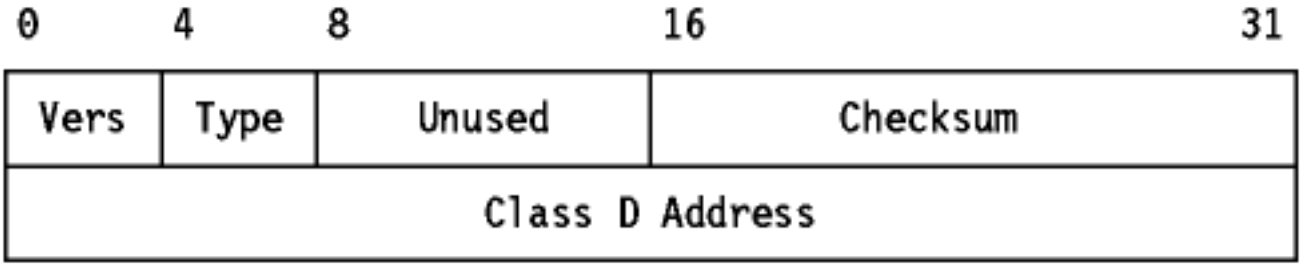


Figura: Formato del mensaje ICMP

Donde:

Vers
: 4 bits que indican la versión IP. Siempre vale 1.

Type
: Especifica que se trata de una consulta o de un informe.

1
: Especifica una consulta enviada por un "router" multicast.

2
: Especifica un informe enviado por un host.

Checksum
: Un checksum de 16 bits calculado de igual forma que con ICMP.

Class D Address
: Es cero para una solicitud, y es una dirección de grupo multicast válida para un informe.

2.7.2 Funcionamiento de IGMP

Los sistemas que participan en IGMP se clasifican en dos tipos: host y "routers" multicast.

Como se describió en Multicasting, con el fin de recibir datagramas de multicast, un host se debe unir a un grupo. Cuando un host es "multi-homed"(multipuerto), puede unirse a cualquier grupo por una o más de sus interfaces(redes a las que está conectado). Los mensajes multicast que recibe el host del mismo grupo en dos subredes distintas pueden ser distintos también. Por ejemplo, 244.0.0.1 es el grupo para "todos los host de esta subred", por lo que los mensajes recibidos en una subred siempre serán diferentes para este grupo que para los de otra subred. Puede haber múltiples procesos en un host a la escucha de mensajes para un grupo de multicast. Si es este el caso, los host se unen sólo una vez al grupo, y llevan la cuenta internamente de qué procesos están interesados en ese grupo.

Para unirse a un grupo, el host envía un informe acerca de una interfaz. El informe se dirige al grupo multicast interesado. Los "routers" multicast de la misma red reciben el informe y activan un flag para indicar que al menos un host de esa red es miembro de ese grupo. Todo host pertenece al grupo 224.0.0.1 de forma automática. Los "routers" multicast tienen que escuchar a todas las direcciones de multicast(todos los grupos) con el fin de detectar tales informes. Las alternativas serían el uso del broadcast para los informes o para configurar host con direcciones unicast para "routers" multicast.

Los "routers" multicast envían regularmente(el RFC 1112 menciona intervalos de 1 minuto) una consulta a la dirección de multicast "todos los hosts". Cada host que aún desee ser miembro de uno o más grupo replica una vez por cada grupo en el que esté interesado(nunca al grupo "todos los hosts", al que pertenece de modo automático). Cada respuesta se envía tras un intervalo de tiempo aleatorio para evitar aglomeraciones en el tráfico. Como a los "routers" no les importa cuántos hosts son miembro de un grupo y como todos los miembros de ese grupo pueden oír las respuestas de cada uno de los demás hosts, cualquier host que escuche a otro host proclamar su pertenencia a mismo grupo cancelará su respuesta para ahorrar recursos. Si ningún host responde dentro de un intervalo de tiempo dado, el "router" multicast decide que ningún host pertenece a ese grupo. Cuando un host o un "router" multicast recibe un datagrama multicast, su acción depende del valor TTL y de la dirección IP de destino.

0
: Un datagrama enviado con un valor TTL de cero se restringe el host emisor.

1
: Un datagrama con un TTL de uno alcanza todos los hosts de la subred que son miembros del grupo. Los "router" multicast decrementan el valor a cero, pero a diferencia de los datagramas unicast, no lo informan con un mensaje ICMP "Time Exceeded". La expiración de un datagrama multicast se considera un evento normal.

2+
: Todos los hosts que sean miembros del grupo y todos los "routers" multicast reciben el datagrama. La acción de los "router" depende de la dirección del grupo multicast.

224.0.0.0 - 224.0.0.255
: Este rango se emplea sólo para aplicaciones multicast que hagan uso de un único salto. Los "routers" multicast no retransmitirán datagramas con direcciones IP en este rango.
A primera vista puede parecer como si un no tuviera que molestarse en informar de la pertenencia a un grupo en este rango ya que los "router" no le retransmitirán los de otras subredes. Sin embargo, el informe indica además a otros host de la subred que el host informante es miembro del grupo. El único grupo del que nunca se da parte es el 224.0.0.1.

other
: El "router" retransmite normalmente los datagramas con otros valores para la dirección de destino: el valor TTL se decrementa en al menos un segundo, como es habitual.
Esto permite a un host localizar al servidor más cercano que esté escuchando sobre un dirección multicast usando lo que se llama *"expanding ring search"(búsqueda expansiva en anillo)*. El host envía un datagrama con un valor TTL de 1(misma subred) y espera una respuesta. Si no se recibe ninguno, prueba con un TTL de 2, luego de 3, etc. Al final encontrará al servidor más cercano. (4)

Tabla de contenidos ARP("Address Resolution Protocol")

2.8 ARP("Address Resolution Protocol")

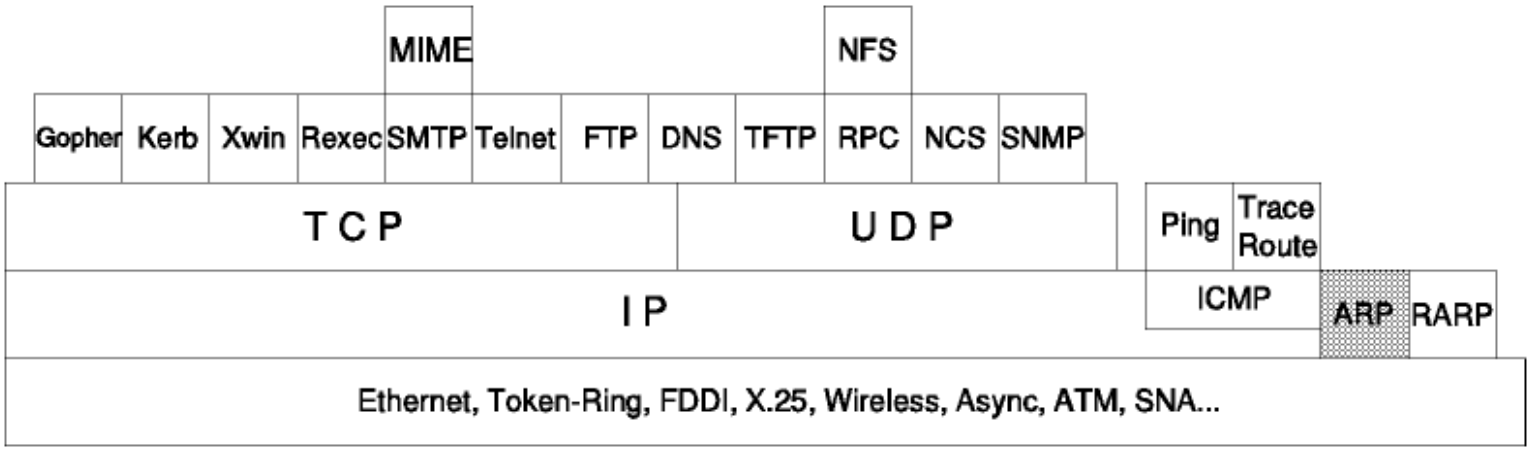


Figura: ARP("Address Resolution Protocol")

El protocolo ARP es un *protocolo estándar específico de las redes*. Su status es *electivo*.

El protocolo de resolución de direcciones es responsable de convertir las dirección de protocolo de alto nivel(direcciones IP) a direcciones de red físicas. Primero, consideremos algunas cuestiones generales acerca de Ethernet.

2.8.1 Ethernet versus IEEE 802.3

Se pueden usar los siguientes formatos de trama en el cable coaxial de Ethernet:

1. El estándar lanzado en 1978 por Xerox Corporation, Intel Corporation y Digital Equipment Corporation, llamado habitualmente *Ethernet* (o Ethernet *DIX*).
2. El estándar internacional IEEE 802.3, definido más recientemente.

La diferencia entre estos dos estándares está en el uso de uno e los campos de la cabecera, que contiene un número de tipo de protocolo para Ethernet y la longitud de los datos del trama en IEEE 802.3.

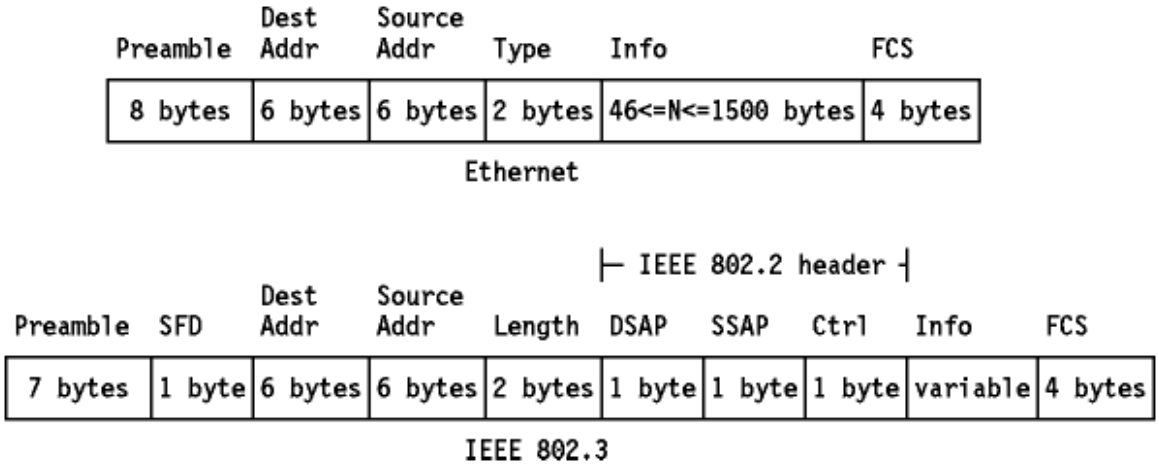


Figura: formatos de trama en Ethernet e IEEE 802.3

- El campo de tipo en Ethernet se usa para distinguir diferentes protocolos ejecutándose en el mismo cable coaxial, y permite su coexistencia en el mismo cable físico.
- La longitud máxima de un trama de Ethernet es de 1526 bytes. Esto significa un campo de datos de hasta 1500 bytes. La longitud del campo de datos en 802.3 está limitada también a 1500 bytes para redes a 10 Mbps, pero es distinta para otras velocidades de transmisión.
- En el trama MAC de 802.3, la longitud del campo de datos la indica la cabecera. El tipo de protocolo figura en la cabecera del protocolo 802.2(de nivel superior), ver Figura - Formatos de trama para Ethernet e IEEE 802.3
- En la práctica, no obstante, ambos formatos de bloque pueden coexistir en un mismo cable físico. Esto se consigue utilizando números de tipo de protocolo(campo de tipo) superiores a 1500 en la trama Ethernet. Sin embargo, es necesario que distintos controladores sean capaces de manejar cada uno de estos formatos.

Así, a efectos prácticos, la capa física de Ethernet e IEEE 802.3 son compatibles. A pesar de todo, las capas de enlace de Ethernet y de IEEE 802.3/802.2 no lo son.

La capa LLC("Logical Link Control") de 802.2

El IEEE 802.3 usa un concepto conocido como *LSAP("Link Service Access Point")* que utiliza una cabecera de 3 bytes:

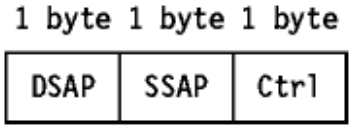


Figura: Cabecera de LSAP en IEEE 802.2

donde DSAP y SSAP significan Destination y Source Service Access Point, respectivamente. Los números para estos campos son asignados por un comité IEEE.

Debido al número creciente de aplicaciones que emplean IEEE 802 como protocolo para los niveles inferiores, se le hizo una extensión en la forma del SNAP ("Sub-Network Access Protocol"). Se trata de una extensión a la cabecera de LSAP, y el valor 170 de los campos SSAP y DSAP indica su uso.

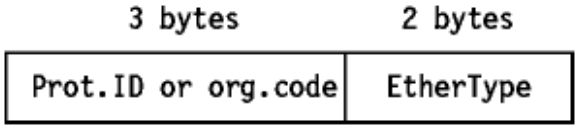


Figura: Cabecera de LSAP en IEEE 802.2

En la evolución de TCP/IP

se han establecido tres estándares, que describen el encapsulamiento de las tramas IP y ARP en estas redes:

1. 1984: *RFC 894 - Estándares para la transmisión de datagramas IP en redes Ethernet* especifica sólo el uso de las redes Ethernet. Los valores asignados al campo de tipo son:

   2048 (hex 0800)
   para datagrama IP
   2054 (hex 0806)
   para datagrama ARP

2. 1985: *RFC 948 - Dos métodos para la transmisión de datagrama IP en redes IEEE 802.3* especifica dos posibilidades:
   1. El método compatible con Ethernet: las tramas se envían en un red IEEE 802.3 como si se tratase de una red Ethernet network, es decir, usando el campo de longitud de IEEE 802.3 como campo de tipo, violando por tanto las reglas de IEEE 802.3.
   2. Formato IEEE 802.2/802.3 LLC tipo 1: usando la cabecera 802.2 LSAP con IP, con el valor 6 para los campos SSAP y DSAP.

   El RFC indica claramente que el método IEEE 802.2/802.3 es el preferido, es decir, se supone que todas las implementaciones futuras de IP en redes IEEE 802.3 deberían usarlo.

3. 1987: *RFC 1010 - Números asignados* (ahora obsoleto por el RFC 1700 de 1994) señala que como resultado de la evolución de IEEE 802.2 y de la necesidad de más números de protocolo, se desarrolló una nueva aproximación al problema basada en el intercambio de experiencias prácticas que tuvo lugar durante la convención de distribuidores de TCP de agosto de 1986. Afirma que de ahora en adelante en todas las implementaciones de IEEE 802.3, 802.4 y 802.5 se debería emplear la versión SNAP("Sub-Network Access Protocol") del IEEE 802.2 LLC: con los campos DSAP y SSAP fijados a 170(indicando el uso de SNAP) y asignando SNAP del modo siguiente:
   - 0 (cero) como código de organización.
   - Campo EtherType:

     2048 (hex 0800)
     para datagrama IP
     2054 (hex 0806)
     para datagrama ARP
     32821 (hex 8035)
     para datagrama RARP

     Estos son los mismos valores que se usan en el campo de tipo de Ethernet.

1. 1988: *RFC 1042 - Estándar para la transmisión de datos en redes IEEE 802*.

   Debido a que este nuevo enfoque del problema(muy importante para las implementaciones) pasó prácticamente inadvertido al figurar como una pequeña nota del RFC, hubo bastante confusión al respecto, y finalmente, en febrero de 1988, se repitió en un RFC propio: *RFC 1042*, que deja obsoleto al *RFC 948*.

Sin embargo, en la práctica, todavía hay implementaciones de TCP/IP que utilizan el viejo método LSAP(RFC 948 o 1042). *Tales implementaciones no se comunicarán con implementaciones más recientes.*

Notar además que el último método cubre no sólo las redes IEEE 802.3, sino también las redes IEEE 802.4 y 802.5.

2.8.2 Descripción de ARP

En una sola red física, los hosts individuales se conocen en la red a través de su dirección física. Los protocolos de alto nivel direccionan a los hosts de destino con una dirección simbólica (en este caso la dirección IP). Cuando tal protocolo quiere enviar un datagrama a la dirección IP de destino w.x.y.z, el manejador de dispositivo no la entiende.

En consecuencia, se suministra un módulo(ARP) que traducirá la dirección IP a las dirección física del host de destino. Utiliza una tabla(llamada a veces *caché ARP*) para realizar esta traducción.

Cuando la dirección no se encuentra en la caché ARP, se envía un broadcast en la red, con un formato especial llamado *petición ARP.* Si una de las máquinas en la red reconoce su propia dirección IP en la petición, devolverá una *respuesta ARP* al host que la solicitó. La respuesta contendrá la dirección física del hardware así como información de encaminamiento8(si el paquete ha atravesado puentes durante su trayecto) tanto esta dirección como la ruta se almacenan en la caché del host solicitante. Todos los posteriores datagramas enviados a esta dirección IP se podrán asociar a la dirección física correspondiente, que será la que utilice el manejador de dispositivo para mandar el datagrama a la red.

ARP se diseño para ser usado en redes que soportasen broadcast por hardware. Esto significa, por ejemplo, que ARP no funcionará en una red X.25.

2.8.3 Concepto detallado de ARP

ARP se emplea en redes IEEE 802 además de en las viejas redes DIX Ethernet para mapear direcciones IP a dirección hardware. Para hacer esto, ha de estar estrechamente relacionado con el manejador de dispositivo de red. De hecho, las especificaciones de ARP en RFC 826 sólo describen su funcionalidad, no su implementación, que depende en gran medida del manejador de dispositivo para el tipo de red correspondiente, que suele estar codificado en el *microcódigo del adaptador*.

2.8.3.1 Generación del paquete ARP

Si una aplicación desea enviar datos a una determinado dirección IP de destino, el mecanismo de encaminamiento IP determina primero la dirección IP del siguiente salto del paquete (que puede ser el propio host de destino o un "router") y el dispositivo hardware al que se debería enviar. Si se trata de una red 802.3/4/5, deberá consultarse el módulo ARP para mapear el par <tipo de protocolo, dirección de destino> a una dirección física.

El módulo ARP intenta hallar la dirección en su caché. Si encuentra el par buscado, devuelve la correspondiente dirección física de 48 bits al llamador(el manejador de dispositivo). Si no lo encuentra, *descarta el paquete* (se asume que al ser un protocolo de alto nivel volverá a transmitirlo) y genera un *broadcast* de red para una solicitud ARP.

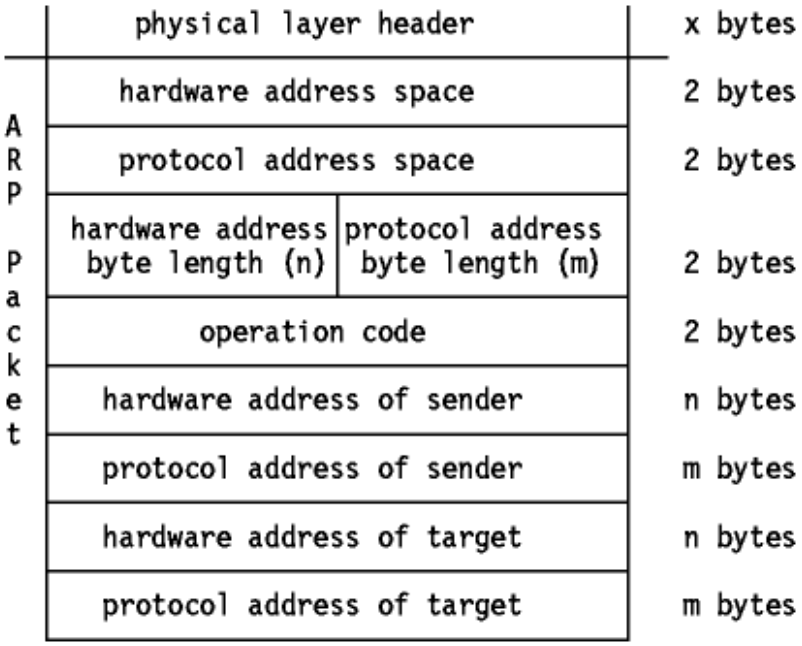


Figura: Paquete de petición/respuesta ARP

Donde:

Hardware address space
: Especifica el tipo de hardware; ejemplos son Ethernet o Packet Radio Net.

Protocol address space
: Especifica el tipo de protocolo, el mismo que en el campo de tipo EtherType en la cabecera de IEEE 802.

Hardware address length
: Especifica la longitud(en bytes) de la dirección hardware del paquete. Para IEEE 802.3 e IEEE 802.5 será de 6.

Protocol address length
: Especifica la longitud(en bytes) de las direcciones del protocolo en el paquete. Para IP será de 4.

Operation code
: Especifica si se trata de una petición(1) o una solicitud(2) ARP.

Source/target hardware address
: Contiene las direcciones física hardware. En IEEE 802.3 son direcciones de 48 bits.

Source/target protocol address
: Contiene las direcciones del protocolo. En TCP/IP son direcciones IP de 32 bits.

Para el paquete de solicitud, la dirección hardware de destino es el único campo indefinido del paquete.

2.8.3.2 Recepción del paquete ARP

Cuando un host recibe un paquete ARP(bien un broadcast o una respuesta punto a punto), el dispositivo receptor le pasa el paquete al módulo ARP, que lo trata como se indica en Figura - Recepción del paquete ARP.

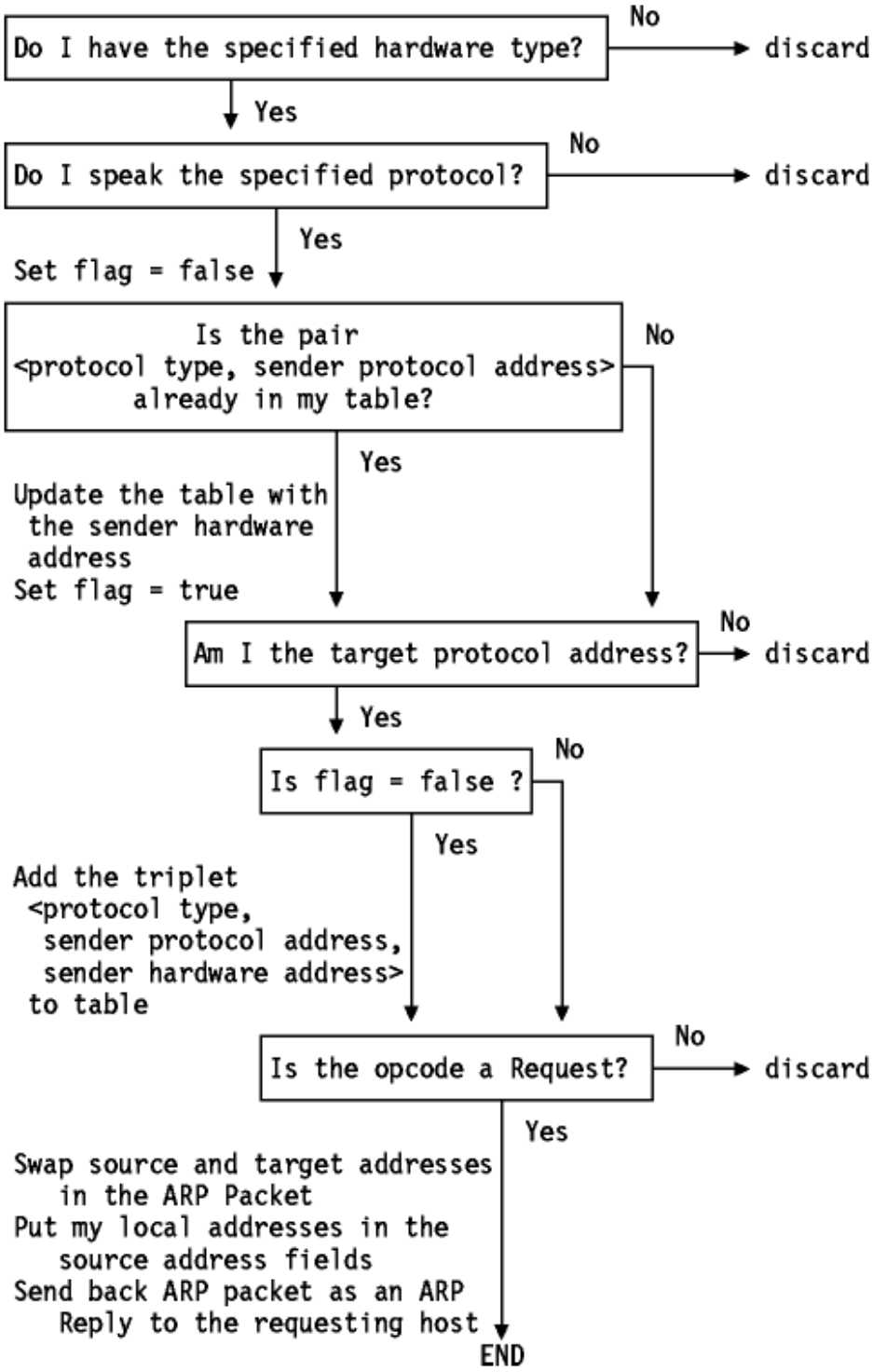


Figura: Recepción del paquete ARP

El host solicitante recibirá esta respuesta ARP, y seguirá el algoritmo ya comentado para tratarla. Como resultado, la tripleta <tipo de protocolo, dirección de protocolo, dirección hardware> para el host en cuestión se añadirá a la caché ARP. La próxima vez que un protocolo de nivel superior quiera enviar un paquete a ese host, el módulo de ARP encontrará la dirección hardware, a la que se enviará el paquete.

Notar que debido a que la petición ARP original fue un broadcast en la red, todos los host en ella habrán actualizado la dirección del emisor en su propia caché(sólo si previamente ya existía esa entrada) en la tabla.

2.8.4 ARP y subredes

El protocolo ARP es el mismo aunque haya subredes. Recordar que cada datagrama IP pasa primero por el algoritmo de encaminamiento IP. Este algoritmo selecciona el manejador de dispositivo que debería enviar el paquete. Sólo entonces se consulta al módulo ARP asociado con ese manejador.

2.8.5 Proxy-ARP o subnetting transparente

El Proxy-ARP se describe en el *RFC 1027 - Usando ARP para implementar pasarelas de subredes transparentes*, que de hecho es un subconjunto del método propuesto en el *RFC 925 - Resolución de direcciones Multi-LAN*. Es otro método para construir subredes locales, sin necesidad de modificar el algoritmo de encaminamiento IP, pero con modificaciones en los routers, que interconectan las subredes.

2.8.5.1 Concepto de Proxy-ARP

Considerar una red IP, dividida en subredes, interconectadas por "routers". Utilizamos el algoritmo IP "viejo", lo que significa que ningún host conoce la existencia de múltiples redes físicas. Si se toman los hosts A y B, que se hallan en distintas redes físicas dentro de la misma red IP, y un "router" entre las dos subredes:

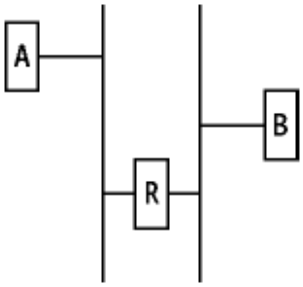


Figura: Hosts interconectados por un "router"

Cuando el host A quiere enviar un datagrama IP al host B, primero ha de determinar la dirección de red física del host B usando ARP.

Como A no puede diferenciar entre las redes físicas, su algoritmo de encaminamiento IP piensa que el host B está en su misma red local y envía un broadcast de petición ARP. El host B no lo recibe, pero sí el "router" R. R entiende de subredes, es decir, ejecuta la versión de subred del algoritmo de encaminamiento y será capaz de ver que el destino de la petición ARP(en el campo de dirección de protocolo de destino) está localizado en otra red física. Si las tablas de encaminamiento de R especifican que el siguiente salto a otra red se produce a través de un dispositivo diferente, replicará al ARP *como si fuera el host B*, diciendo que la dirección de B es la del mismo "router".

El host A recibe esta respuesta ARP, la introduce en su caché y enviará los siguientes paquetes dirigidos a B al "router" R, que los retransmitirá a la subred adecuada.

El resultado es subnetting transparente:

- Los host normales(como A y B) desconocen el subnetting, por lo que usan el algoritmo de encaminamiento clásico.
- Los "router" entre subredes":
    1. Utilizan el algoritmo IP para subredes.
    2. Usan un módulo ARP modificado, que puede responder en nombre de otros hosts.

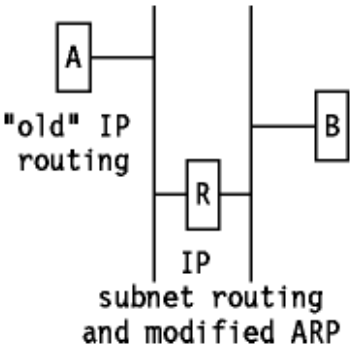


Figura: "Router" Proxy-ARP

Tabla de contenidos RARP ("Reverse Address Resolution Protocol")

Tabla de contenidos Concepto de Proxy-ARP

2.9 RARP("Reverse Address Resolution Protocol")

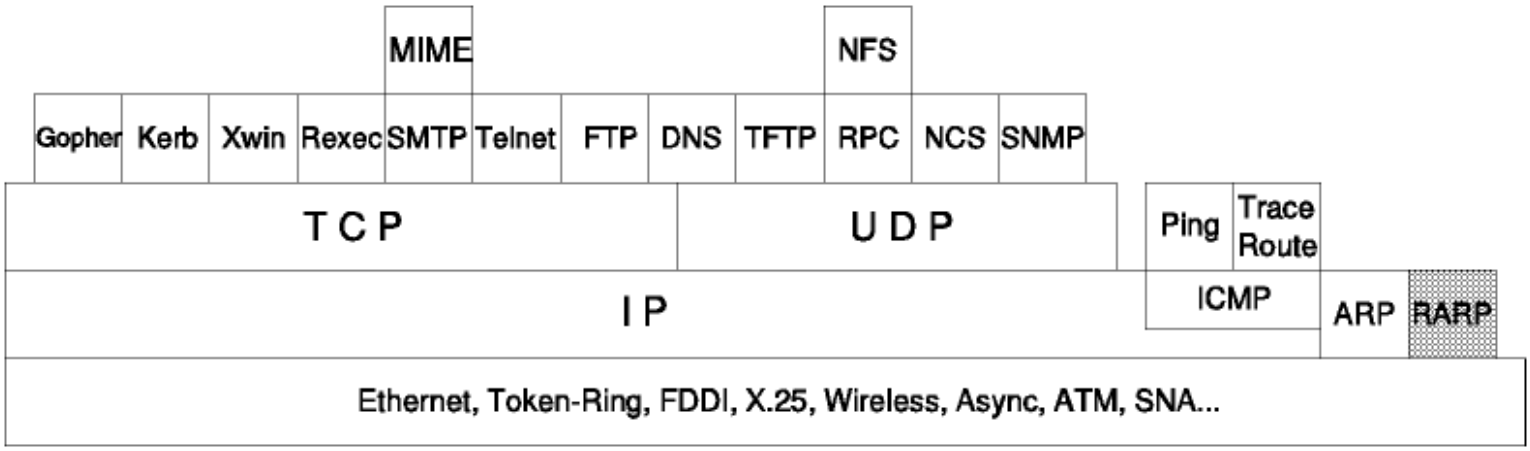


Figura: RARP("Reverse Address Resolution Protocol")

2.9.1 Descripción de RARP

El protocolo ARP es un *protocolo estándar específico de redes*. Su status es *electivo*.

Algunos hosts, como por ejemplo estaciones de trabajo sin disco, desconocen su propia dirección IP cuando arrancan. Para determinarla, emplean un mecanismo similar al ARP, pero ahora el parámetro conocido es la dirección hardware el host y el requerido su dirección IP. La diferencia básica con ARP es el hecho de que debe existir un "servidor RARP" en la red que mantenga una base de datos de mapeados de direcciones hardware a direcciones de protocolo.

2.9.2 Concepto de RARP

El cálculo de direcciones inversas se efectúa del mismo modo que en ARP. Se usa el mismo formato de paquete(ver Figura - Paquete petición/respuesta de ARP).

Una excepción es el campo "operation code" que ahora toma los siguientes valores:

3
: para la petición RARP

4
: para la respuesta RARP

Y, por supuesto, cabecera "física" de la trama indicará ahora que RARP es el protocolo de nivel superior (8035 hex) en vez de ARP (0806 hex) o IP (0800 hex) en el campo *EtherType*. El mismo concepto de RARP genera algunas diferencias:

- ARP asume sólo que cada host conoce el mapeado entre su propia dirección hardware y de protocolo. RARP requiere uno o más hosts en la red para mantener una base de datos con los mapeados entre direcciones de red direcciones de protocolo de modo que serán capaces de responder a solicitudes de los host clientes.
- Debido al tamaño que puede tomar esta base de datos, parte de las funciones del servidor suelen implementarse fuera del microcódigo del adaptador, con la opción de una pequeña caché en el microcódigo, que sólo es responsable de la recepción y transmisión de tramas RARP, estando el mapeado RARP en sí a cargo del software que se ejecuta en el servidor como un proceso normal.
- La naturaleza de esta base de datos también requiere algún software para crear y actualizar la base de datos manualmente.
- En caso de que haya múltiples servidores RARP en la red, el cliente RARP sólo hará uso de la primera respuesta RARP que reciba a su broadcast, y desechará las otras.

Tabla de contenidos Puertos y zócalos

2.10 Puertos y zócalos

En esta sección se presentan los conceptos de *puertos* y *zócalos*.

2.10.1 Puertos

Cada proceso que se desea comunicar con otro se identifica en la pila de protocolos TCP/IP con uno o más puertos. Un puerto es un número de 16 bits, empleado por un protocolo host – a – host para identificar a que protocolo del nivel superior o programa de aplicación se deben entregar los mensajes recibidos.

Como algunas aplicaciones son ya de por sí protocolos estandarizados, como TELNET y FTP, emplean el mismo número de puerto en todas las implementaciones TCP/IP. Estos puertos "asignados" se conocen como *puertos bien conocidos*, y a sus aplicaciones, *aplicaciones bien conocidas*.

Estos puertos son controlados y asignados por IANA ("Internet Assigned Numbers Authority") y en la mayoría de los sistemas sólo los puede utilizar los procesos del sistema o los programas que ejecutan usuarios privilegiados. Ocupan número de puerto comprendidos en el rango de 0 a 1023. Los puertos con números en el rango de 1024 a 65535 no los controla IANA y en la mayor parte de los sistemas los pueden usar los programas de usuario.

La confusión que se produce cuando dos aplicaciones distintas intentan usar los mismos puertos en un host se evita haciendo que soliciten un puerto disponible a TCP/IP. Como este número se asigna dinámicamente, puede ser diferente en cada ejecución de una misma aplicación.

UDP, TCP y ISO TP-4 están todos basados en el mismo principio de uso de los puertos (Ver Figure - UDP, A Demultiplexer Based on Ports y Figure - TCP Connection.) En la medida de lo posible, se usan los mismos números para los servicios situados sobre UDP, TCP y ISO TP-4.

2.10.2 Zócalos

En primer lugar, conviene definir los siguientes términos:

- Un *zócalo* es un tipo especial de *descriptor de fichero* que un proceso usa para solicitar servicios de red al sistema operativo.
- Una dirección de zócalos es la tripleta:

  {*protocolo, dirección local, proceso local* }

  En la pila TCP/IP, por ejemplo:

  {*tcp, 193.44.234.3, 12345*}

- Una *conversación* es el enlace de comunicaciones entre dos procesos.
- Una *asociación* es la quíntupla que especifica completamente los dos procesos comprendidos en una conexión:

  {*protocolo, dirección local, proceso - local, dirección exterior, proceso exterior*}

  En la pila TCP/IP, por ejemplo:

  {*tcp, 193.44.234.3, 1500, 193.44.234.5, 21*}

  podría ser una asociación válida.

- Una *medio – asociación* es:

{*protocolo, dirección - local, proceso local* }

o

{*protocolo, dirección exterior, proceso exterior*}

que especifican cada una de las mitades de la conexión.

- La *medio – asociación* se denomina también zócalo o *dirección de transporte*. Es decir, un zócalo es un punto terminal para la comunicación que puede ser nombrado y direccionado en una red.

La interfaz del zócalo es una de tantas APIs con los protocolos de comunicación. Se introdujo por primera vez en el UNIX BSDD 4.2. Aunque no ha sido estandarizada, se ha convertido en un estándar *de facto*.

4.2BSD permitía dos dominios de comunicación distintos: Internet y UNIX. 4.3BSD ha añadido los protocolos del XNS ("Xerox Network System") y 4.4BSD añadirá una interfaz extendida para soportar los protocolos OSI.

2.10.3 Llamadas básicas de zócalos

A continuación se muestran algunas llamadas básicas de la interfaz de zócalos.

- Inicializar un zócalo

  FORMAT: int *sockfd* = **socket**(int *family*, int *type*, int *protocol*)

  donde:

  - *family* es la *familia de direccionamiento*. Puede tomar valores como AF_UNIX, AF_INET, AF_NS y AF_IUCV. Su fin es especificar el método de direccionamiento que usa el zócalo.
  - *type* es el tipo de interfaz de zócalo a usar. Puede tomar valores como SOCK_STREAM, SOCK_DGRAM, SOCK_RAW, y SOCK_SEQPACKET.
  - *protocol* puede ser UDP, TCP, IP o ICMP.
  - *sockfd* es un entero (similar a un descriptor de fichero) devuelto por la llamada a **socket**.
- Registrar un zócalo en una dirección de puerto

  FORMAT: int **bind**(int *sockfd*, struct sockaddr **localaddr*, int *addrlen*)

  donde:

  - *sockfd* wa wl mismo entero que devuelve la llamada a **socket**.
  - *localaddr* es la dirección local que devuelve la llamada a **bind**.

  Nótese que tras la llamada a **bind**, ya hay valores para los tres primeros parámetros de la asociación:

  {*protocol, local-address, local-process, foreign-address, foreign-process*}

- Indica disponibilidad para recibir conexiones

  FORMAT: int **listen**(int *sockfd*, int *queue-size*)

  donde:

  - *sockfd* es el mismo entero que devuelve la llamada a **socket**.
  - *queue-size* indica el número de solicitudes de conexión que se pueden encolar en el sistema mientras el proceso local no ha llamado todavía a **accept**.
- Acepta una conexión

FORMAT: int **accept**(int *sockfd*, struct sockaddr **foreign-address*, int *addrlen*)

donde:

- *sockfd* es el mismo entero que devuelve la llamada a **socket**.
- *foreign-address* es la dirección del proceso cliente que devuelve la llamada a **accept**.

Nótese que la llamada a **accept** la efectúa un proceso servidor más que un cliente. Si hay una solicitud de conexión encolada, **accept** toma la primera solicitud de la cola y crea otro zócalo con las mismas propiedades que *sockfd*; en otro caso, **accept** bloquea el llamador hasta que llega una solicitud de conexión.

- Solicita la conexión con el servidor

FORMAT: int **connect**(int *sockfd*, struct sockaddr **foreign-address*, int *addrlen*)

donde:

- *sockfd* es el mismo entero que devuelve la llamada a **socket**.
- *foreign-address* es la dirección del cliente que devuelve la llamada a **connect**.

Nótese que esta llamada la efectúa un proceso cliente más que uno servidor.

- Envía o recibe datos

Las funciones **read**(), **readv**(*sockfd*, char **buffer*, int *addrlen*), **recv**(), **readfrom**(), **send**(*sockfd, msg, len, flags*), **write**() se pueden emplear para recibir y enviar datos en una asociación ya establecida.

Nótese que estas llamadas son similares a las llamadas estándar del sistema de E/S **read** y **write**.

- Cierra un zócalo

FORMAT: int **close**(int *sockfd*)

donde:

- *sockfd* es el mismo entero que devuelve la llamada a **socket**.

2.10.4 Ejemplo

Como ejemplo, considérese las llamadas del sistema de zócalos para un protocolo orientado a conexión.

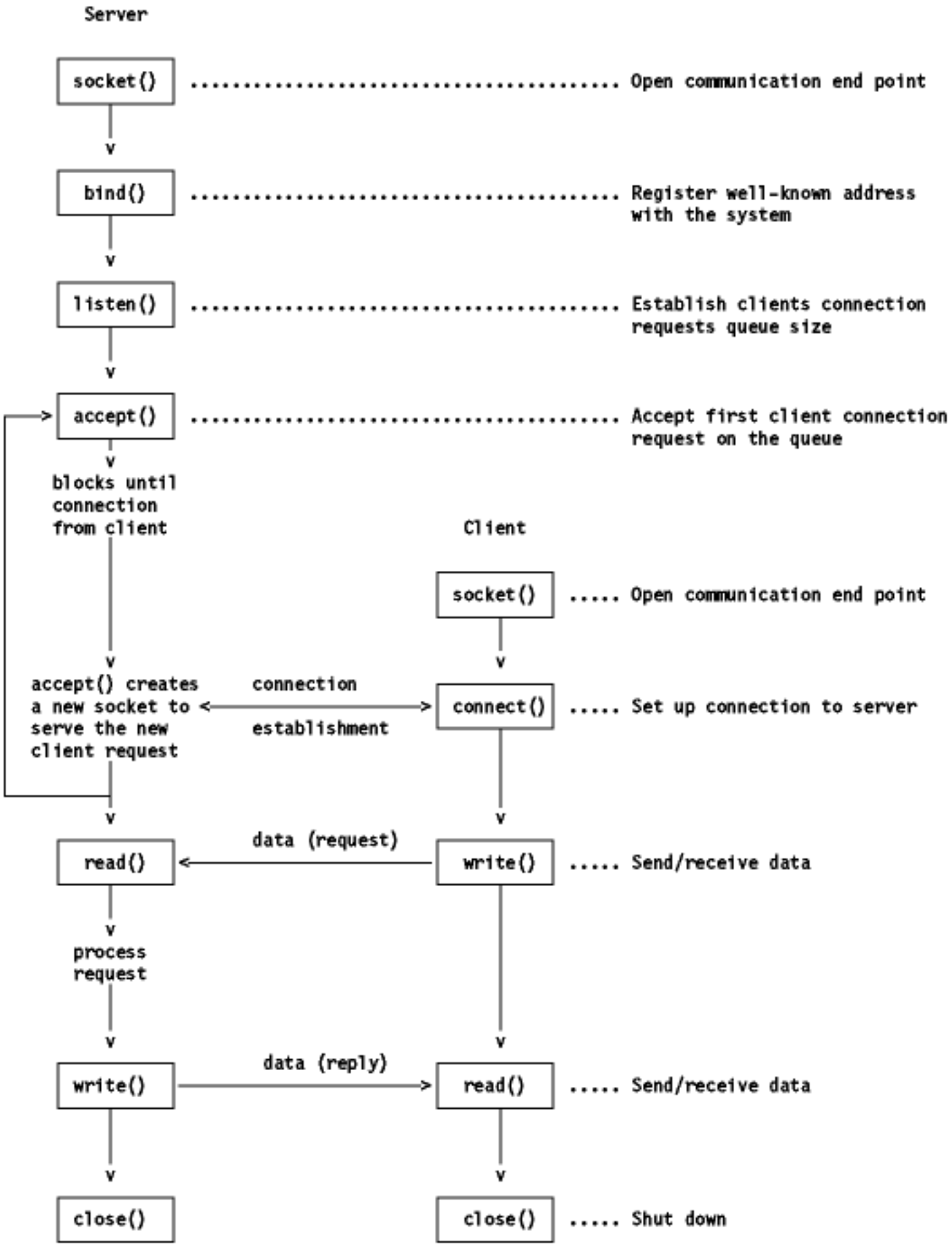


Figura: Llamadas del sistema para zócalos para un protocolo orientado a conexión

Especificando los elementos de la asociación:

	Protocol	Local Address , Local Process	Foreign Address , Foreign Process
connection-oriented server	socket()	bind()	listen() accept()
connection-oriented client	socket()	connect()	
connectionless server	socket()	bind()	recvfrom()
connectionless client	socket()	bind()	sendto()

Figura: Llamadas del sistema para zócalos y asociación

La interfaz de zócalos se distingue por los diferentes servicios suministrados. Flujo, datagramas y zócalos a bajo nivel definen cada uno distintos servicios disponibles para las aplicaciones.

- **Interfaz de zócalos orientada flujo** (SOCK_STREAM): Define una conexión fiable en un servicio orientado a conexión. Los datos se envían sin errores y sin duplicados y se reciben en el mismo orden en el que se envían. El control de flujo está integrado para evitar el desbordamiento de datos. No se imponen límites sobre los datos intercambiados, que se consideran un flujo de bytes. Un ejemplo de aplicación que usa esta interfaz es el FTP.
- **Interfaz de zócalos orientada a datagramas** (SOCK_DGRAM): Define un servicio no orientado a conexión (sobre UDP, por ejemplo). Los datagramas se envían como paquetes independientes. El servicio no proporciona garantías; los datos se pueden perder o duplicar, y los datagramas pueden llegar fuera de orden. No se realiza ningún tipo de ensamblaje o desensamblaje de los paquetes. Un ejemplo de aplicación que usa esta interfaz es el NFS ("Network File System").
- **Interfaz de zócalos a bajo nivel** (SOCK_RAW): Permite acceso directo a protocolos de bajo nivel tales como IP e ICMP. Esta interfaz suele usarse para probar implementaciones de nuevos protocolos. Un ejemplo de aplicación que usa esta interfaz es el comando Ping.

Table of Contents UDP(User Datagram Protocol)

2.11 UDP(User Datagram Protocol)

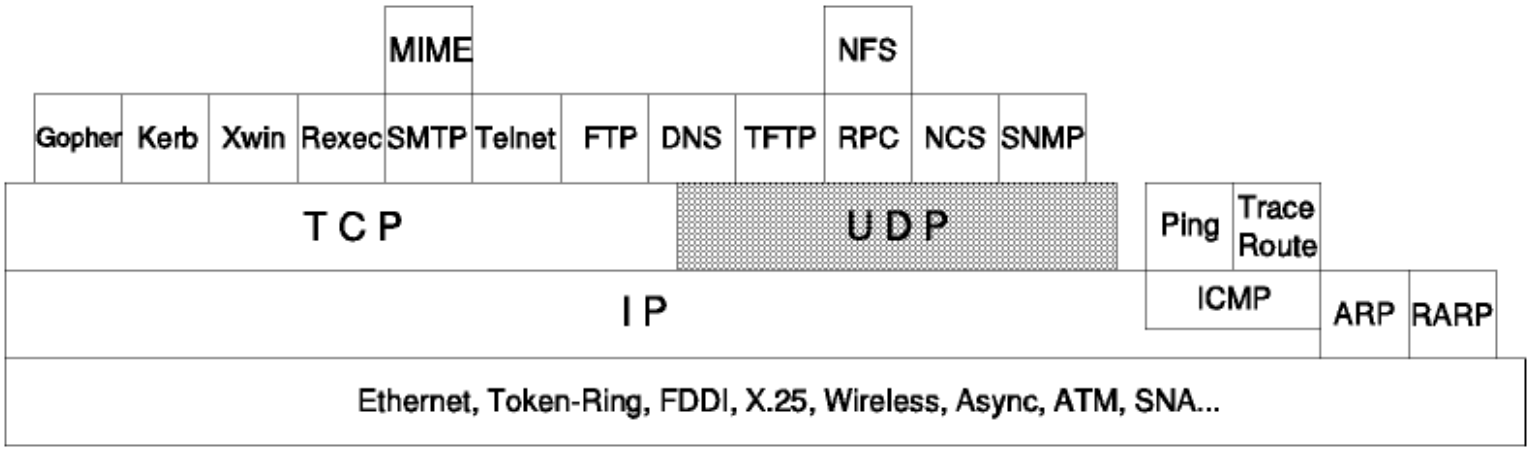


Figura: UDP(User Datagram Protocol)

UDP es un *protocolo estándar*, y su número STD es 6. El *RFC 768 - "User Datagram Protocol"* describe UDP. Su status es *recomendado*, pero en la práctica cualquier implementación de TCP/IP que no se use exclusivamente para el encaminamiento incluye UDP.

Para IP, UDP es básicamente un interfaz de aplicación. No añade fiabilidad, control de flujo
o recuperación de errores a IP. Simplemente sirve como "multiplexor/ demultiplexor" para enviar y recibir datagramas, usando *los puertos* para dirigir los datagramas tal como se muestra en Figura - UDP, un demultiplexor basado en puertos. Para un estudio más detallado de los puertos, remitirse a Puertos y zócalos.

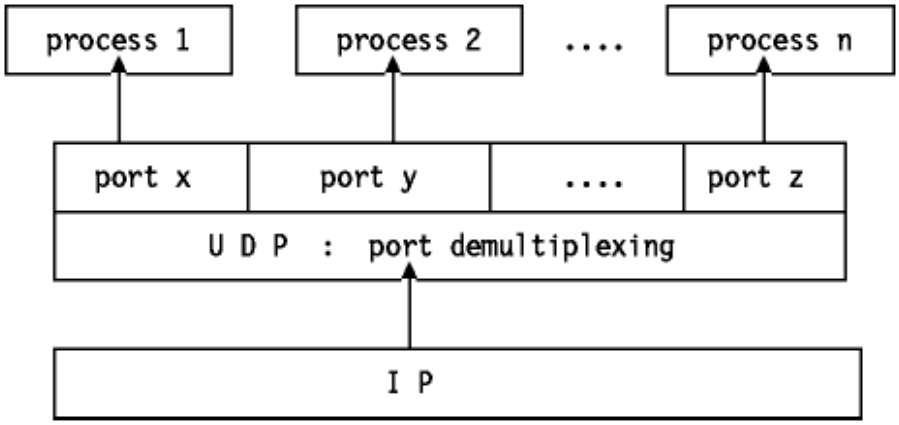


Figura: UDP, un demultiplexor basado en puertos

UDP suministra un mecanismo para que una aplicación envíe un datagrama a otra. Se considera que la capa de UDP es extremadamente delgada y en consecuencia tiene poco "overhead", pero requiere que la aplicación se responsabilice de la recuperación de errores y todo lo que ello conlleva.

2.11.1 Puertos

El concepto de puerto ha sido analizado anteriormente en Puertos y zócalos.

Las aplicaciones que envían datagramas a un host necesitan identificar un objetivo más específico que la dirección IP, ya que los datagramas suelen dirigirse a procesos concretos y no a todo el sistema. UDP permite hacer esto al hacer uso de los *puertos*.

Un puerto es un número de 16 bits que identifica en un host que proceso está asociado a un datagrama. Hay dos tipos de puerto:

Bien-conocidos("well-known").
: Los puertos bien-conocidos pertenecen a servidores estándar, por ejemplo Telnet usa el puerto 23. Los puertos bien-conocidos se hallan en el rango de 1 a 1023 (anteriormente a 1992, el rango de 256 a 1023 se usaba para servidores específicos de UNIX). Estos puertos suelen tener números impares, debido a que los primeros sistemas que usaron el concepto de puerto requerían para las operaciones en duplex una pareja par/impar de puertos. La mayoría de los servidores requieren sólo un único puerto. Una excepción es el servidor BOOTP que usa dos: el 67 y el 68 (ver "BOOTstrap Protocol" - BOOTP).

  La razón de ser de los puertos bien-conocidos es permitir a los clientes encontrar a los servidores sin necesidad de información de configuración. Los números de los puertos bien-conocidos se definen en *STD 2 - Números asignados de Internet("Assigned Internet Numbers")*.

Efímeros
: Los clientes no necesitan puertos bien-conocidos porque inician la comunicación con los servidores y los datagramas UDP enviados al servido contienen su número de puerto. El host en funcionamiento proporciona un puerto a cada proceso cliente mientras este lo necesite. Los números de puertos efímeros tienen valores mayores de 1023, por lo general en el rango de 1024 a 5000. Un cliente puede usar cualquier número en ese rango, siempre que la combinación <protocolo de transporte, dirección IP, número de puerto> sea unívoca .

Nota: TCP también usa puertos con los mismos valores. Estos puertos son totalmente independientes de los de UDP. Normalmente, un servidor usará TCP o UDP, aunque hay excepciones. Por ejemplo, el DNS(ver DNS("Domain Name System")) usa tanto el puerto 53 de UDP como el 53 de TCP.

2.11.2 Formato del datagrama UDP

Cada datagrama UDP se envía en un sólo datagrama de IP. Aunque el datagrama IP se fragmente durante la transmisión, la implementación de IP que lo reciba lo reensamblará antes de pasárselo a la capa de UDP. Todas las implementaciones de IP deben aceptar datagramas de 576 bytes, lo que significa que si se supone un tamaño máximo de 60 bytes para la cabecera IP, queda un tamaño de 516 bytes para el datagrama UDP, aceptado por todas las implementaciones. Muchas implementaciones aceptan datagramas más grandes, pero no es algo que esté garantizado. El datagrama UDP tiene una cabecera de 16 bytes que se describe en Figura - Formato del datagrama UDP.

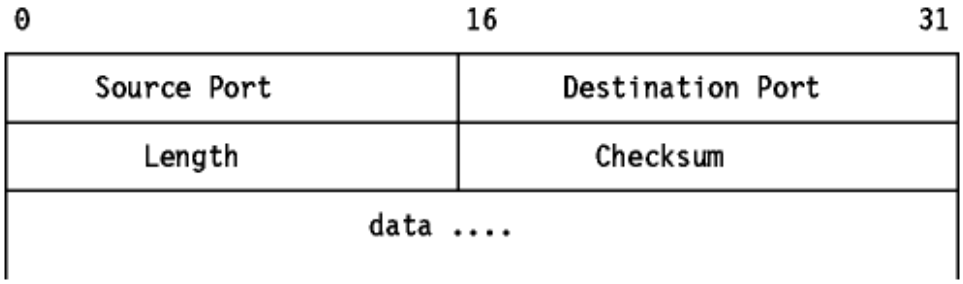


Figura: Formato del datagrama UDP

Donde:

Puerto origen:
: Indica el puerto del proceso que envía el datagrama. Es el puerto al que se deberían dirigir las respuestas.

Puerto destino:
: Especifica el puerto destino en el host de destino.

Longitud
: Es la longitud(en bytes) del mismo datagrama de usuario, incluyendo la cabecera.

Checksum
: Es un campo opcional consistente en el complemento a uno de 16 bits de la suma en complemento a uno de una pseudocabecera IP, la cabecera UDP y los datos del datagrama UDP. La pseudocabecera IP contiene las direcciones IP de origen y destino, el protocolo y la longitud del datagrama UDP:

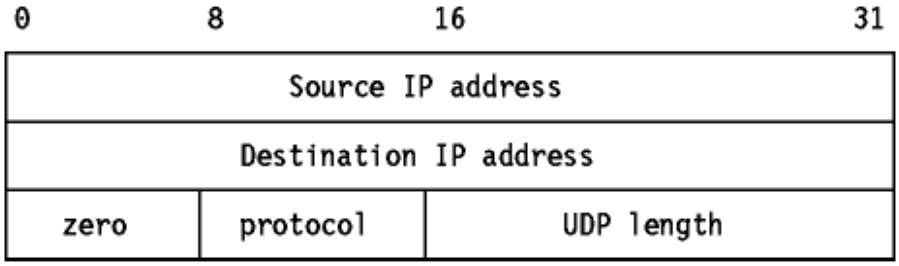


Figura: Pseudocabecera IP

La pseudocabecera IP extiende de modo efectivo su checksum que incluya al datagrama IP original(sin fragmentar).

2.11.3 Interfaz de Programación de Aplicación(API) de UDP

La API que ofrecida UDP se describe en el en RFC 768. Proporciona:

- La creación de nuevos puertos para la recepción.
- Operación de recepción que devuelve los bytes de datos recibidos y una indicación del puerto y la dirección IP de origen.
- Operación de envío que tiene como parámetros los datos, los puertos de origen y destino y las direcciones IP.

La forma en que se implementa esto queda a elección del cada distribuidor.

Hay que ser consciente de que IP y UDP no proporcionan una entrega garantizada, control de flujo ni recuperación de errores, así que estos deberán ser implementados por la aplicación.

Aplicaciones estándar que usan UDP son:

- TFTP("Trivial File Transfer Protocol")
- DNS("Domain Name System")
- RPC("Remote Procedure Call"), usado por el NFS("Network File System")
- NCS("Network Computing System")
- SNMP("Simple Network Management Protocol")

Tabla de contenidos TCP ("Transmission Control Protocol")

2.12 TCP("Transmission Control Protocol")

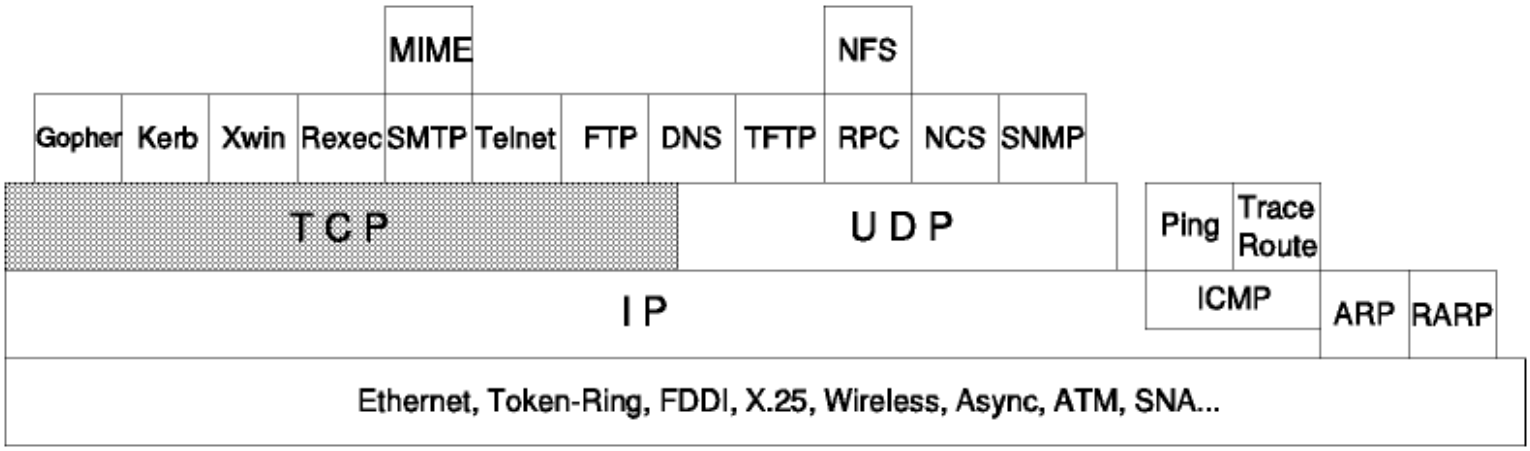


Figura: TCP("Transmission Control Protocol")

TCP es un *protocolo estándar* con el STD 7. Se describe en el *RFC 793 - TCP("Transmission Control Protocol")*. Su status es *recomendado*, pero en la práctica cualquier implementación de TCP/IP que no se use exclusivamente para el encaminamiento incluirá TCP.

TCP proporciona una cantidad considerablemente mayor de servicios a las aplicaciones que UDP, notablemente, la recuperación de errores, control de flujo y fiabilidad. Se trata de un protocolo *orientado a conexión* a diferencia de UDP. La mayoría de los protocolo de aplicación de usuario, como TELNET y FTP, usan TCP.

2.12.1 Zócalos

El concepto de zócalo ya se ha discutido en Puertos y zócalos.

Dos procesos se comunican a través de *zócalos TCP*. El modelo de zócalo proporciona a un proceso una conexión con un flujo full duplex de bytes con otro proceso. La aplicación no necesita preocuparse de la gestión de este canal; estos servicios son suministrados por TCP.

TCP usa el mismo principio de puerto que UDP(ver Puertos) para conseguir multiplexación. Al igual que UDP, TCP utiliza puertos efímeros y bien conocidos. Cada extremo de una conexión TCP tiene un *zócalo* que puede identificarse con la tripleta <TCP, dirección IP address, número de puerto>. Es lo que se llama una *medio asociación*. Si dos procesos se están comunicando sobre TCP, tendrán una *conexión lógica* identificable unívocamente por medio de los dos zócalos implicados, es decir, con la combinación <TCP, dirección IP local, puerto local, dirección IP remota, puerto remoto>. Ver Figura - Conexión TCP. Los procesos del servidor son capaces de gestionar múltiples conversaciones a través de un único puerto.

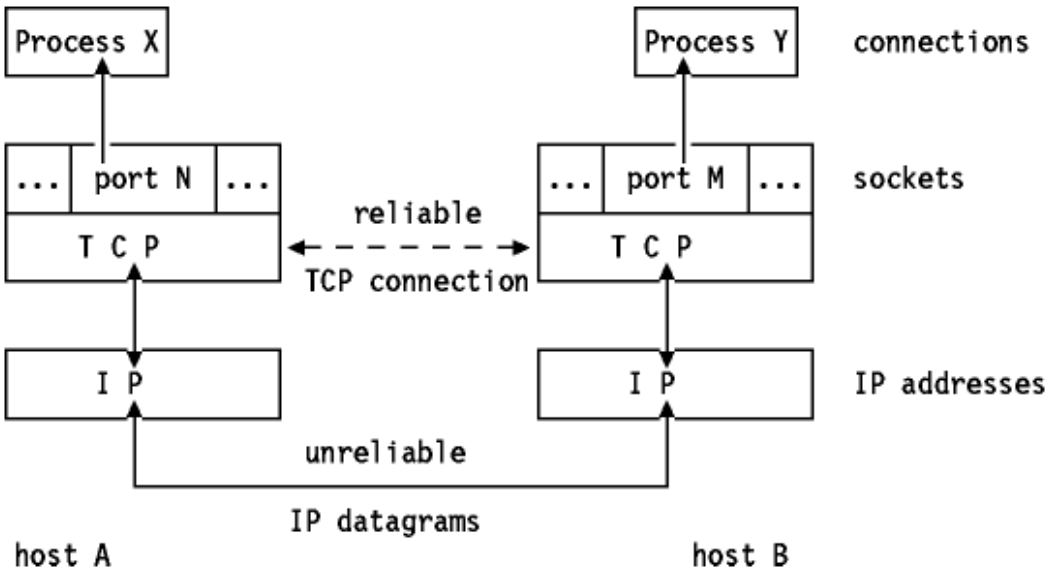


Figura: Conexión TCP - Los procesos X e Y se comunican sobre una conexión TCP que emplea datagramas IP.

2.12.2 Concepto de TCP

Como se señaló arriba, el principal propósito de TCP es *proporcionar una conexión lógica fiable entre parejas procesos. No* asume la fiabilidad de los protocolos de niveles inferiores(como IP) por lo que debe ocuparse de garantizarla.

TCP se puede caracterizar por los siguientes servicios que suministra a las aplicaciones que lo usan:

Transferencia de datos a través de un canal
: Desde el punto de vista de la aplicación, TCP transfiere *un flujo continuo de bytes* a través de Internet. La aplicación no ha de preocuparse de trocear los datos en bloques o en datagramas. TCP se encarga de esto al agrupar los bytes en *segmentos TCP*, que se pasan a IP para ser retransmitidos al destino. Además, TCP decide por sí mismo

cómo segmentar los datos y puede enviarlos del modo que más le convenga.
A veces, una aplicación necesita estar segura de que todos los datos pasados a TCP han sido transmitidos efectivamente al destino. Por esa razón, se define la función *"push"*. Esta función mandará todos los segmentos que sigan almacenados al host de destino. El *cierre normal de la conexión* también provoca que se llame a esta función, para evitar que la transmisión quede incompleta.

Fiabilidad
: TCP asigna un número de secuencia a cada byte transmitido, y espera una reconocimiento afirmativo(ACK) del TCP receptor. Si el ACK no se recibe dentro de un intervalo de timeout, los datos se retransmiten. Como los datos se transmiten en bloques(segmentos de TCP), al host de destino sólo se le envía el número de secuencia del byte de cada segmento.
El TCP receptor utiliza los números de secuencia para organizar los segmentos cuando llegan fuera de orden, así como para eliminar segmentos duplicados.

Control de flujo
: El TCP receptor, al enviar un ACK al emisor, indica también el número de bytes que puede recibir aún, sin que se produzca sobrecarga y desbordamiento de sus buffers internos. Este valor se envía en el ACK en la forma del número de secuencia más elevado que se puede recibir sin problemas. Este mecanismo se conoce también como mecanismo de *ventanas* y se estudiará posteriormente en este capítulo.

Multiplexación
: Se consigue usando puertos, al igual que en UDP.

Conexiones lógicas
: La fiabilidad y el control de flujo descritos más arriba requieren que TCP inicialice y mantenga cierta información de estado para cada canal. La combinación de este estado, incluyendo zócalos, números de secuencia y tamaños de ventanas, se denomina conexión lógica. Cada conexión se identifica unívocamente por el par de zócalos del emisor y el receptor.

Full Duplex
: TCP garantiza la concurrencia de los flujos de datos en ambos sentidos e la conexión.

2.12.2.1 El principio de la ventana

Un simple protocolo de transporte podría emplear el siguiente principio: enviar un paquete, y esperar un reconocimiento del receptor antes de enviar el siguiente. Si el ACK no se recibe dentro de cierto límite de tiempo, se retransmite.

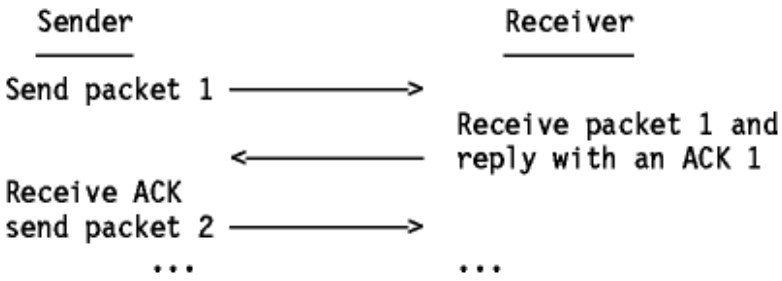


Figura: El principio de la ventana

Aunque este mecanismo asegura fiabilidad, sólo usa una parte del *ancho de banda de la red* que está disponible.

Considerar ahora un protocolo en el que el emisor agrupa los paquetes que va a transmitir como se muestra en Figura - Paquetes del mensaje:

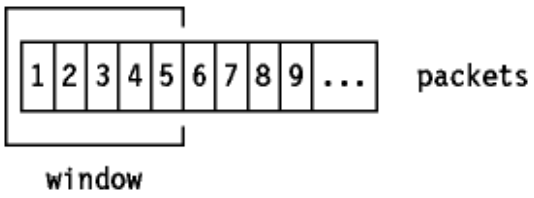


Figura: Paquetes del mensaje

Y utiliza las siguientes reglas:

- El emisor puede enviar todos los paquetes dentro de la ventana sin recibir un ACK, pero debe disparar un cronómetro para el timeout para cada uno de ellos.
- El receptor debe reconocer cada paquete recibido, indicando el número de secuencia del último paquete bien recibido.
- El emisor desliza la ventana para cada ACK recibido.

En nuestro ejemplo, el emisor puede transmitir paquetes del 1 al 5 sin esperar respuesta:

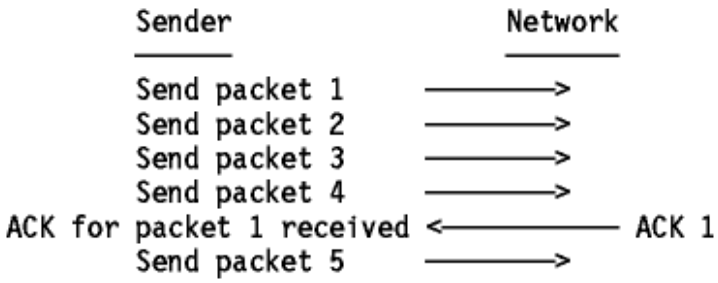


Figura: El principio de la ventana

En el momento en que el emisor recibe el ACK 1, puede deslizar su ventana para excluir el paquete 1:

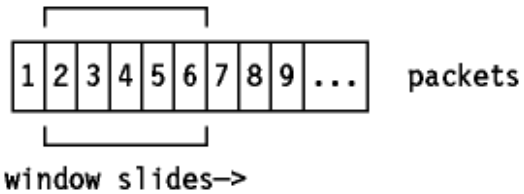


Figura: Paquetes del mensaje

En este punto, el emisor puede transmitir también el paquete 6.

Imaginar algunos casos especiales:

- El paquete 2 se pierde: el emisor no recibirá ACK 2, por lo que su ventana permanecerá en posición 1(como se ve en el último dibujo). De hecho, como el receptor no recibió el paquete 2, reconocerá los paquetes 3, 4 y 5 con un ACK 1, que fueron los últimos paquetes recibidos en secuencia. En el extremo del emisor, al final se producirá un timeout para el paquete 2 y se retransmitirá. Notar que la recepción de este paquete en el receptor generará un ACK 5, ya que se habrán recibido con éxito los paquetes del 1 al 5, y la ventana del emisor se deslizará cuatro posiciones al recibir el ACK 5.
- El paquete 2 llegó, pero el reconocimiento se perdió: el emisor no recibe ACK 2, pero recibe ACK 3. ACK 3 es un reconocimiento de *todos* los paquetes hasta el 3(incluyendo el 2) y el emisor ya puede deslizar su ventana hasta el paquete 4.

Conclusión:

Este mecanismo de ventanas asegura:

- Transmisión fiable
- Mejor aprovechamiento del ancho de banda(mejora del flujo).
- Control de flujo, ya que el receptor puede retrasar la respuesta a un paquete con un reconocimiento, conociendo los buffers libres de los que dispone y el tamaño de la ventana de comunicación.

2.12.2.2 El principio de la ventana aplicado a TCP

El mecanismo mostrado más arriba se utiliza en TCP, pero con unas cuantas diferencias:

- Como TCP proporciona una conexión con un flujo de bytes, los números de secuencia se asignan a cada byte del canal. TCP divide el flujo de bytes en segmentos. El principio de la ventana se aplica a nivel de bytes; es decir, los segmentos enviados y los ACKs recibidos llevarán números de secuencia de forma que el tamaño de la ventana se exprese con un número de bytes, en vez del de paquetes.
- El tamaño de la ventana lo determina el receptor, cuando se establece la conexión, y puede *variar* durante la transmisión de datos. Cada ACK incluirá el tamaño de la ventana que acepta el receptor en ese momento.

Ahora, el flujo de datos del emisor se puede ver como:

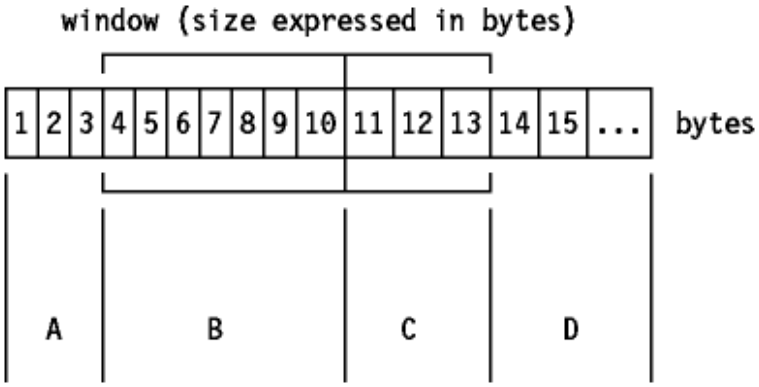


Figura: Principio de la ventana aplicado a TCP

Donde:

A
: Bytes transmitidos que han sido reconocidos.

B
: Bytes enviados pero no reconocidos.

C
: Bytes que se pueden enviar sin esperar ningún tipo de reconocimiento.

D
: Bytes que no se pueden enviar aún.

Recordar que TCP agrupa los bytes en segmentos, y un segmento TCP sólo lleva el número de secuencia del primer byte.

2.12.2.3 Formato de segmento en TCP

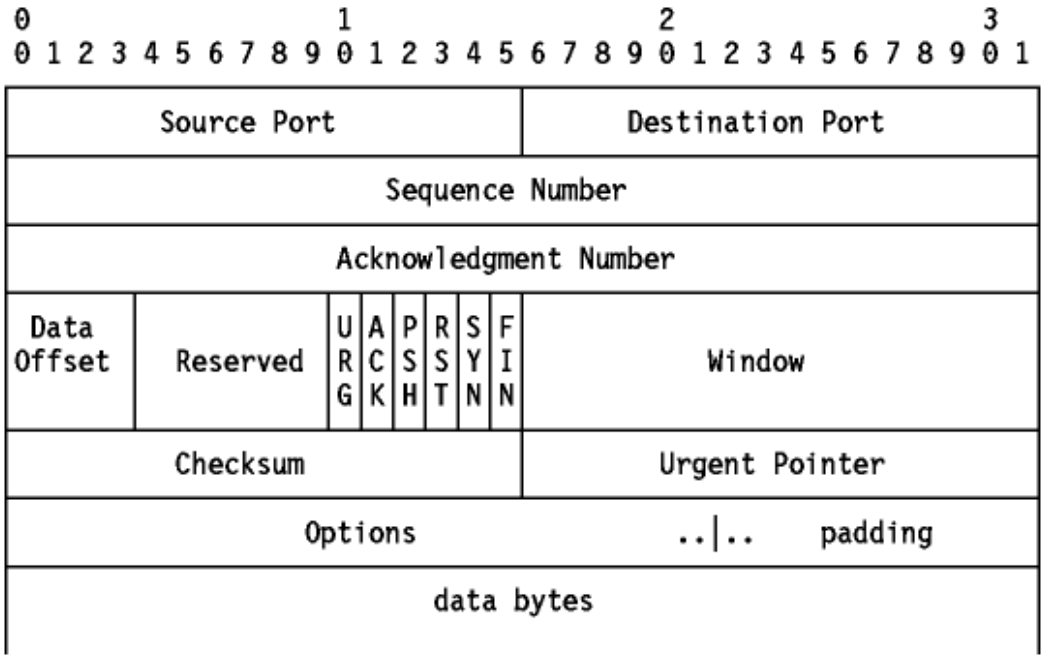


Figura: Formato de segmento en TCP

Donde:

Source Port
: El número de puerto de 16 bits del emisor, que el receptor usa para responder.

Destination Port
: El número de puerto de 16 bits del receptor.

Sequence Number
: El número de secuencia del primer byte de datos del segmento. Si el byte de control SYN está a 1, el número de secuencia es el inicial(n) y el primer byte de datos será el n+1.

Acknowledgment Number
: Si el bit de control ACK está a 1, este campo contiene el valor del siguiente número de secuencia que se espera recibir.

Data Offset
: El número de palabras de 32 bits de la cabecera TCP. Indica dónde empiezan los datos.

Reserved
: Seis bits reservados para su uso futuro; deben ser cero.

URG
: Indica que el campo "urgent pointer" es significativo en el segmento.

ACK
: Indica que el campo de reconocimiento es significativo en el segmento.

PSH
: Función "Push".

RST
: Resetea la conexión.

SYN
: Sincroniza los números de secuencia.

FIN
: No hay más datos del emisor.

Window
: Usado en segmentos ACK. Especifica el número de bytes de datos que comienzan con el byte indicado en el campo número de reconocimiento que el receptor esta dispuesto a aceptar.

Checksum
: El complemento a uno de 16 bits de la suma de los complementos a uno de todas las palabras de 16 bits de la pseudocabecera, la cabecera TCP y los datos TCP. Al computar el checksum, el mismo campo checksum se considera cero.
La pseudocabecera es la misma que utiliza UDP para calcular el checksum. Es una pseudocabecera IP, usada sólo para calcular el checksum, con el formato mostrado en Figura - Pseudocabecera IP:

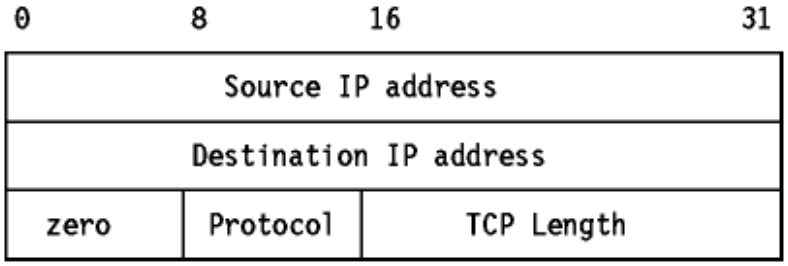


Figura: Pseudocabecera IP

Urgent Pointer
: Apunta al primer octeto de datos que sigue a los datos importantes. Sólo es significativo cuando el bit de control URG está a uno.

Options
: Sólo para el caso de opciones de datagramas IP, las opciones pueden ser:
- Un sólo byte conteniendo el número de opción, o
- Una opción de longitud variable con el siguiente formato:

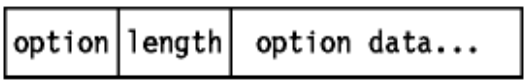


Figura: Opción del datagrama IP - Opción de longitud variable.

Actualmente hay definidas tres opciones:

Tipo	Longitud	Significado
0	-	Fin e la lista de opciones.
1	-	No-Operación.
2	4	Tamaño máximo del segmento.

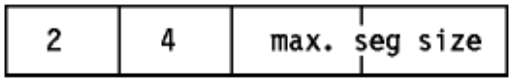


Figura: Opción tamaño máximo del segmento

Esta opción sólo se usa durante el establecimiento de la conexión (bit de control SYN puesto a uno) y se envía desde el extremo que ha de recibir datos para indicar la máxima longitud de segmento que es capaz de manejar. Si esta opción no se usa, se admiten segmentos de cualquier tamaño.

Padding

Bytes todos a cero para rellenar la cabecera TCP a una longitud total que sea un múltiplo de 32 bits.

2.12.2.4 Reconocimientos y retransmisiones

TCP envía los datos en segmentos de longitud variable. Los números de secuencia se basan en una cuenta de los bytes. Los *reconocimientos especifican el número de secuencia del siguiente byte que el receptor espera recibir*.

Ahora suponer que un segmento se pierde o se corrompe. En ese caso, el receptor reconocerá cualquier segmento sucesivo con un reconocimiento referido al primer byte del paquete perdido. Finalmente, se producirá un timeout y el segmento perdido se retransmitirá.

Suponer un tamaño de ventana de 1500 bytes, y segmentos de 500 bytes.

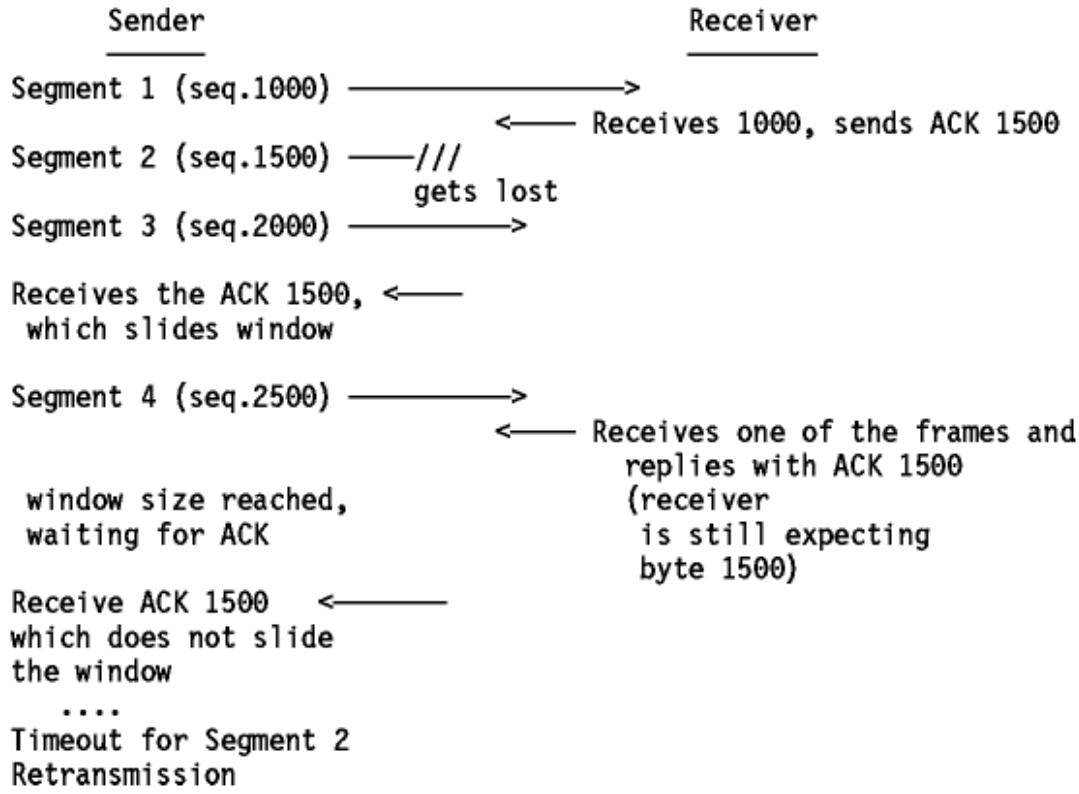


Figura: Proceso de reconocimiento y retransmisión

Ahora surge un problema, ya que el emisor sabe que el segmento 2 está perdido o corrompido, pero no sabe nada de los segmentos 3 y 4. El emisor debería retransmitir al menos el segmento 2, pero también podría retransmitir los segmentos 3 y4. Es posible que:

1. El segmento 3 haya sido recibido, y no se sepa nada del 4: podría haber sido recibido ya, sin que el ACK haya llegado, o se podría haber perdido también.
2. El segmento 3 se haya perdido, y se haya recibido el ACK 1500 a la recepción del segmento 4.

Cada implementación de TCP es libre de reaccionar ante un timeout del modo que deseen los diseñadores. Podría retransmitir sólo el segmento 2, pero en el segundo caso indicado arriba, estaremos esperando hasta que el timeout del segmento 3 expire. En este caso, se pierden todas las ventajas del rendimiento del mecanismo de ventanas. O bien TCP podría reenviar inmediatamente todos los segmentos de la ventana actual.

Sea cual sea la elección, el rendimiento máximo se pierde. Esto se debe a que el ACK no contiene un segundo número de secuencia indicando la trama actual que se ha recibido.

Intervalos de timeout variable

Cada TCP debería implementar un algoritmo para adaptar los tiempos de timeout a usar para el viaje de los segmentos. Para hacerlo, TCP registra el momento de envío de un segmento, y el de recepción del ACK. Se promedia un valor para varios de estos viajes que se empleará como valor de timeout para el siguiente segmento a enviar.

Esto es una característica importante, ya que los retardos pueden ser variables en la red, dependiendo de múltiples factores, tales como la carga de las redes intermedias de baja

velocidad o la saturación de las pasarelas.

2.12.2.5 Estableciendo una conexión TCP

Antes de que se pueda transferir cualquier dato, se ha de establecer una conexión entre los dos procesos. Uno de los procesos(normalmente el servidor) lanza una llamada *OPEN pasiva*, el otro una llamada *OPEN activa*. El OPEN pasivo permanece dormido hasta que otro proceso intenta comunicarse con él a través de un OPEN activo.

En la red, se intercambian tres segmentos TCP:

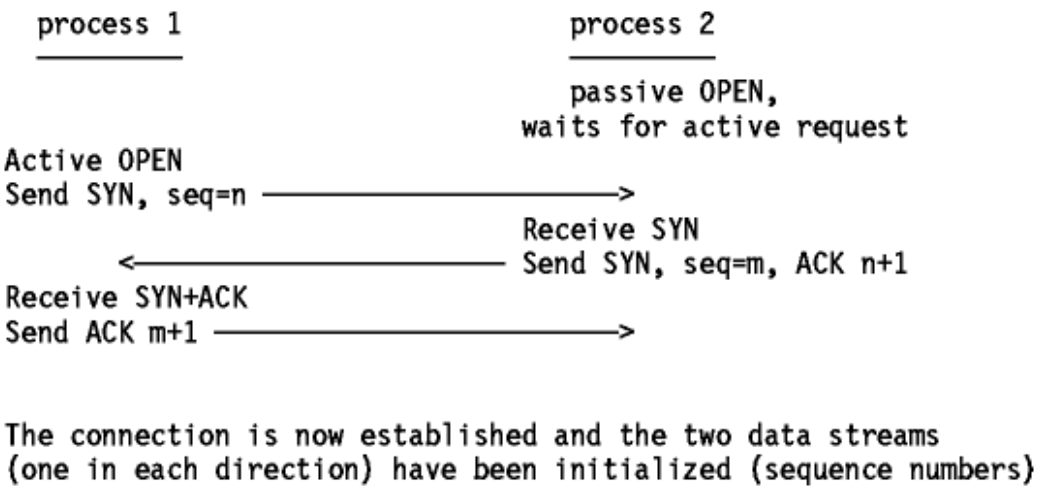


Figura: Establecimiento de la conexión TCP

Este proceso completo se conoce como *three-way handshake*, o acuerdo en tres fases. Notar que los segmentos TCP intercambiados incluyen los números de secuencia iniciales de ambas partes, para ser usados en posteriores transferencias.

El *cierre* de la conexión se hace de forma implícita enviando un segmento TCP con el bit FIN activo. Como la conexión es full duplex, el segmento FIN sólo cierra la conexión en un sentido del canal. El otro proceso enviará los datos restantes, seguidos de un segmento TCP en el que el bit FIN está activo. La conexión se borra(es decir, la información de estado en ambos extremos) una vez que el canal se ha cerrado en ambos sentidos.

2.12.2.6 Segmentos TCP transportados en datagramas IP

Los TCP segmentos se transportan sobre datagramas IP con la siguiente configuración de parámetros:

```
Tipo de servicio = 00000000
  es decir: precedencia = rutina
            retraso = normal
            rendimiento = normal

TTL = 00111100 (1 minute)
```

2.12.3 API de TCP

La API de TCP no está definida del todo. Sólo algunas funciones básicas que deberían ser proporcionadas se describen en el *RFC 793 - TCP("Transmission Control Protocol")*. Como ocurre con la mayoría de los RFCs de la pila de protocolos TCP/IP, se deja un elevado grado de libertad a los diseñadores, permitiendo en consecuencia implementaciones óptimas (dependientes del sistema operativo), lo que resulta en una mayor eficiencia.

El RFC describe las siguientes llamadas a funciones:

Open
: Para establecer una conexión, tiene varios parámetros:
  - Activo/pasivo
  - Zócalo remoto
  - Número de puerto local
  - Timeout(opcional)
  - Y muchas otras opciones.

  Devuelve un *nombre para la conexión local*, que se usa para referenciarla en todas las otras funciones.

Send
: Hace que los datos del buffer del usuario señalado se envíen por la conexión. Opcionalmente puede tener los flags URGENT o PUSH activo.

Receive
: Copia los datos TCP que van llegando a un buffer de usuario.

Close
: Cierra la conexión; provoca un "push" de todos los restantes datos y segmentos TCP con el flag FIN activo.

Status
: Es una llamada dependiente de la implementación que devuelve información como:
  - Zócalo local y remoto
  - Tamaños de las ventanas de recepción y envío
  - Estado de la conexión
  - Nombre de la conexión local

Abort
: Hace que todas las operaciones de recepción y envío aborten, y se envíe un RESET al TCP remoto.

Tabla de contenidos ATM ("Asynchronous Transfer Mode")

2.13 ATM("Asynchronous Transfer Mode")

Las redes basadas en ATM están teniendo un creciente interés para las aplicaciones, tanto LAN como WAN. Ya hay algunos productos disponibles para construir una red física ATM propia. La arquitectura ATM es nueva y distinta de las arquitecturas LAN estándar. Por este motivo, son necesarios cambios para que los productos LAN tradicionales funcionen en entornos ATM. En el caso de TCP/IP, el principal cambio está en la interfaz de red para que soporte ATM.

Existen ya varios enfoques del problema, dos de los cuales son importantes para el transporte del tráfico TCP/IP. Se describen en IP clásico sobre ATM y Emulación LAN con ATM. También se comparan en IP clásico sobre ATM versus emulación LAN.

2.13.1 Resolución de direcciones(ATMARP y InATMARP)

La resolución de direcciones en una subred lógica IP de ATM la hace el ATMARP ("ATM Address Resolution Protocol") basado en el RFC 826 y en el InATMARP("Inverse ATM Address Resolution Protocol") basado en el RFC 1293. ATMARP es el mismo protocolo que ARP pero con las extensiones necesarias para que ARP funcione en el entorno de servidor unicast de ATM. InATMARP es el mismo protocolo que el InARP original, pero aplicado a redes ATM. El uso de estos protocolos difiere en si se utilizan o no PVCs o SVCs.

Tanto ATMARP como InATMARP están definidos en el RFC 1577, que es una propuesta de estándar con estado electivo.

La encapsulación de las peticiones/respuestas de InATMARP y ATMARP se describe en IP clásico sobre ATM.

2.13.1.1 InATMARP

El protocolo ARP se usa para calcular la dirección hardware de un host a partir de su dirección IP. El protocolo InATMARP se usa para calcular la dirección IP de un host a partir de su dirección hardware. En un entorno conmutado, primero se establece una VC("Virtual Connection") o una PVC("Permanent Virtual Connection") o una SVC("Switched Virtual Connection") para comunicar con otra estación. Por lo tanto, se sabe la dirección hardware del otro host, pero no la dirección IP. InATMARP proporciona resolución dinámica de direcciones. Utiliza el mismo formato de trama que el ARP estándar , pero define dos nuevos códigos de operación:

- InARP request=8
- InARP reply=9

Ver Generación del paquete ARP para más detalles.

El InATMARP básico opera esencialmente del mismo modo que ARP, con la excepción de que no hace las peticiones con broadcasts. Esto se debe a que la dirección hardware ya se conoce. Una estación solicitante simplemente formatea una petición insertando sus direcciones hardware de IP(fuente) y la dirección hardware del destino. Luego rellena con ceros el campo de dirección IP del destino y envía el mensaje a la estación de destino. Para cada petición InATMARP, la estación receptora formatea una respuesta utilizando la dirección fuente de la petición como dirección de destino para la respuesta. Ambos extremos actualizan sus tablas ARP. El valor *tipo de hardware* para ATM es el 19 decimal y el campo *EtherType* se pone a 0x806, que indica ARP según el RFC 1700.

2.13.1.2 Resolución de dirección en el entorno PVC

En un entorno PVC cada estación utiliza el protocolo InATMARP para determinar las direcciones IP de todas las demás estaciones conectadas. La resolución se hace para aquellos PVCs configurados para la encapsulación LLC/SNAP. Es responsabilidad de cada estación IP que soporte PVCs la revalidación de las entradas de la tabla ARP a medida que pasa el tiempo.

2.13.1.3 Resolución de direcciones en el entorno SVC

SVCs requiere soporte para ATMARP en el entorno no-broadcast de ATM. Para hacer frente a esta necesidad, se debe localizar

un único servidor ATMARP dentro de la LIS("Logical IP Subnetwork"; ver LIS("Logical IP Subnetwork")). Este servidor tiene la responsabilidad de resolver las peticiones ATMARP de todos los miembros IP del LIS. Para una explicación de los términos ATM, remitirse a IP clásico sobre ATM.

El servidor en sí mismo no establece conexiones de modo activo. El inicio del proceso de registro ATMARP depende el cliente del LIS. Un cliente individual se conecta al servidor ATMARP con una conexión punto a punto VC. El servidor, al completarse la conexión ATM de un nuevo VC con encapsulación LLC/SNAP, transmitirá un petición InATMARP para determinar la dirección IP del cliente. La respuesta InATMARP del cliente contiene la información necesaria para que el servidor construya su caché ATMARP. Esta tabla consiste en:

- dirección IP
- dirección ATM
- Sello de tiempo("Timestamp")
- VC asociado

Esta información se usa para generar respuestas a las peticiones ATMARP recibidas.

Nota: El servidor ATMARP requiere que cada cliente sea configurado administrativamente con la dirección ATM del servidor ATMARP.

Algoritmo de inserción/actualización de la tabla ARP:

- Si el servidor ATMARP recibe una nueva dirección IP en una respuesta InATMARP la dirección IP se añade a la tabla ATMARP.
- Si la dirección IP de la respuesta InATMARP duplica una dirección IP de una entrada de la tabla y la dirección InATMARP de ATM no coincide la dirección ATM de esa entrada en la tabla y existe un VC abierto asociado a esa entrada, la información InATMARP se desecha y no se hacen cambios en la tabla.
- Cuando el servidor recibe una petición ATMARP sobre un VC, en el que la dirección IP y ATM de la fuente coinciden con la asociación que ya existe en la tabla, y la dirección ATM coincide con la que está asociada al VC, el servidor actualiza el timeout de la entrada en su tabla para la fuente. Por ejemplo, si el cliente está enviando solicitudes ATMARP al servidor sobre el mismo VC usado para registrarse, el servidor se da cuenta de que ese cliente sigue "vivo" y actualiza su timeout en la tabla.
- Cuando el servidor recibe un ARP_REQUEST sobre un VC, examina la fuente de la información. Si no hay ninguna dirección IP asociada a ese VC y si la dirección IP de la fuente no está asociada a ninguna otra conexión, entonces el servidor añade esa estación a su tabla. Este no es el procedimiento normal ya que, como se indica arriba, es responsabilidad del cliente registrarse en el servidor ATMARP.

Degeneración de la tabla ATMARP

Las entradas de la tabla ATMARP son válidas:

- En clientes por un máximo de 15 minutos
- En servidor por una mínimo de 20 minutos

Antes de invalidar una entrada de su tabla, el servidor ATPARP genera un InARP_REQUEST para cualquier VC abierto asociado con esa entrada y decide lo que ha de hacer de acuerdo con las siguientes reglas:

- Si se recibe una respuesta InARP_REPLY, la entrada en la tabla se actualiza en vez de borrarse.
- Si no hay ningún VC asociado a esa entrada, la entrada se borra.

Por tanto, si el cliente no mantiene un VC abierto al servidor, debe refrescar su información ATMARP en el servidor al menos cada 20 minutos. Esto se hace abriendo un VC al servidor de intercambiando los paquetes InATMARP iniciales.

El cliente maneja las actualizaciones de la tabla con el siguiente criterio:

- Cuando una entrada de la tabla degenera, el cliente la invalida.
- Si no hay un VC asociado a la entrada invalidada, se borra.
- En el caso de una entrada invalidada con un VC abierto, el cliente ATMARP revalida la entrada para ese VC antes de enviar cualquier información que no tenga nada que ver con la resolución de direcciones. Hay dos posibilidades:
    - En el caso de un PVC, el cliente valida la entrada al transmitir un InARP_REQUEST y actualizar la entrada al

recibir un InARP_REPLY.
    - En el caso de un SVC, el cliente valida la entrada al transmitir un ARP_REQUEST al servidor ATMARP y actualizar la entrada al recibir un ARP_REPLY.
- Si un VC asociado con una entrada invalidada de la tabla ATMARP se cierra, la entrada se elimina.

Como se menciona arriba, cualquier cliente IP de ATM que use SVCs debe conocer la dirección de su servidor ATM para el LIS concreto. Esta dirección se le debe indicar a cada cliente durante la configuración. Por el momento no hay ninguna dirección ATMARP bien conocida.

2.13.2 IP clásico sobre ATM

Las definiciones de implementaciones de IP clásico sobre ATM se describen en el RFC 1577, que es una propuesta de estándar con status electivo según el RFC 1720(STD 1). Este RFC considera sólo la aplicación de ATM como una sustitución directa de los "cables"("wires"), segmentos LAN locales que conectan estaciones IP como extremos de la conexión("members") y "routers" que operan sobre el paradigma LAN clásico. Las consecuencias derivadas de los puentes a nivel MAC y de la emulación LAN no se tienen en cuenta.

Una distribución inicial de ATM proporciona una sustitución de los segmentos LAN por:

- Ethernets, redes en anillo o FDDI
- Troncales de área local entre LANs ya existentes(no ATM)
- Circuitos dedicados a PVCs por retransmisión de tramas("Frame Relay") entre "router" IP

Este RFC describe también extensiones al ARP(RFC 826). Este tema se discute aparte en Resolución de direcciones(InATMARP y ATMARP).

Algunos fundamentos de ATM:

Celdas
: Todo tipo de información(voz, imágenes, vídeo, datos, etc.) se transporta a través de la red en bloques muy pequeños(48 bytes de datos más una cabecera de 5 bytes) llamados celdas.

Encaminamiento
: El flujo de información se produce a lo largo de rutas(llamadas "canales virtuales") establecidas como una serie de punteros por la red. La cabecera de una celda contiene un identificador que vincula la celda al camino correcto que debe tomar para llegar a su destino.
Las celdas de un canal virtual particular siempre siguen el mismo camino y se entregan en el destino en el mismo orden en el que llegaron al canal.

Conmutación por hardware
: ATM está diseñado de tal forma que se emplean simples elementos de lógica hardware en cada nodo para realizar la conmutación. En un enlace de 1 Gbps llega una nueva celda, y se transmite una celda cada 0.43 microsegundos. El tiempo de conmutación es mínimo.

Conexión virtual VC
: ATM proporciona un entorno conmutado VC("Virtual Connection"). El VC se puede establecer bien a partir de un PVC("Permanent Virtual Connection") o de un SVC("Switched Virtual Connection") dinámico. La gestión de SVC hace con implementaciones del protocolo Q.93B.

Interfaz de usuario final
: La única forma para que un protocolo de nivel superior se comunique sobre una red ATM es por medio de la capa ATM AAL("ATM Adaptation Layer"). La función de esta capa es realizar el mapeado entre las PDUs y las celdas. Hay cuatro tipos diferentes de AAL, AAL1, AAL2, AAL3/4 Y AAL5. Estos AALs ofrecen distintos servicios a los protocolos de nivel superior. Aquí se muestran las características de AAL5, usado para TCP/IP:
    - Modo mensaje y modo flujo
    - Entrega garantizada
    - Entrega no garantizada(usada por TCP/IP)
    - Fragmentación de los datos en bloques y segmentos
    - Operación multipunto

  AAl5 proporciona las mismas funciones que una LAN en el nivel MAC("Medium Access Control"). Los extremos del VC saben el tipo de AAL por medio del mecanismo de configuración de la celda, por lo que la cabecera de la celda no ha de llevarlo. Para los PVCs el tipo AAL se configura administrativamente en los extremos cuando se establece la conexión. ara los SVCs, el tipo de AAL se comunica por el canal vía 0.93B como parte de la solicitud de establecimiento y definición de la conexión y los extremos usan las señales de control para configurarse. Los conmutadores ATM no suelen preocuparse del tipo de AAL de los VCs. El formato AAL5 especifica un formato de paquete con un tamaño

máximo de 64KB - 1 byte de usuario. Las "primitivas" que ha de usar el protocolo de nivel superior como interfaz con la capa AAL(en el SAP de AAL("Service Access Point") están definidas rigurosamente. Cuando un protocolo de nivel superior envía datos, estos son procesados primero por la capa de adaptación, luego por ATM y por último la capa física se encarga de enviar los datos por la red ATM. Las celdas se transportan y las recibe el otro extremo de la conexión en su capa física, que las pasa a ATM, que tras procesarlas las pasa al AAL receptor, que a su vez devuelve los datos al nivel superior. La función total que ha realizado la red ATM ha sido un transporte no garantizado de información(se podría haber perdido una parte). Desde un punto de vista más conservador del proceso de datos, todo lo que ha hecho la red ATM ha sido sustituir un enlace físico por otro tipo de conexión física - todos los protocolos de alto nivel siguen teniendo que efectuarse(por ejemplo IEEE 802.2).

Direccionamiento

Una dirección ATM de un extremo de la conexión se codifica bien como una dirección de 20 bytes basada en OSI NSAP(utilizada para direccionamiento en redes privadas, con tres formatos posibles) o como una dirección E.164 Public UNI(del estilo de los números telefónicos, usados para redes TM públicas).(5)

Broadcast, Multicast

En la actualidad no hay funciones de broadcast similares a las de las LANs. Pero sí existe una función de multicast. El término ATM para multicast es "conexión punto - multipunto".

2.13.2.1 LIS("Logical IP Subnetwork")

El término LIS se introdujo para mapear la estructura lógica de IP a la red ATM. En el contexto LIS, cada entidad administrativa independiente configura sus hosts y "router" dentro de una red IP. Cada LIS opera y se comunica con independencia de otros LIS de la misma red ATM. Los host conectados a una red ATM se comunican directamente con otros hosts dentro del mismo LIS. Esto implica que todos los miembros de un LIS sean capaces de comunicarse con otros hosts del mismo LIS por medio de ATM. La comunicación con hosts externos al propio LIS requiere un "router". El "router" es un extremo ATM conectado a la red ATM que se configura como un miembro de uno o más LISs. Esta configuración puede dar lugar a un número de LISs distintos operando sobre la misma red ATM. Los hosts de diferentes subredes deben usar un "router" aunque se pueda abrir un VC entre ellos a través de ATM.

2.13.2.2 Encapsulación multiprotocolo

Si se desea usar más de un tipo de protocolo de red(IP, IPX, etc.) concurrentemente en una red física, se necesita una forma de multiplexar los distintos protocolos. Esto se puede hacer en el caso de ATM con una multiplexación basada en VC o en LLC. Si se elige multiplexación por VC, hay que tener un VC para cada protocolo entre los dos hosts. La encapsulación LLC proporciona la función de multiplexación en la capa LLC y por tanto sólo requiere un VC. TCP/IP usa, según los RFCs 1577 y 1483, el segundo método debido a que esta forma de multiplexado ya estaba definida en el RFC 1042 para otros tipos de LAN como Ethernet, FDDI y redes en anillo. Con esta definición, IP simplemente usa ATM en vez de una LAN. Todos los demás beneficios que ofrece ATM, como el transporte de tráfico isócrono, etc., no se utilizan. Hay un grupo IETF trabajando en la mejora de la implementación de IP y su interfaz con ATM.

Para ser precisos, la PDU de TCP/IP se encapsula en una cabecera IEEE 802.2 LLC seguida de una cabecera IEEE 802.1a SNAP("SubNetwork Attachment Point") transportada dentro del campo de carga útil("payload field") de una PDU AAL5 CPCS("Common Part Convergence Sublayer"). El formato de la PDU AAL5 CPCS se muestra en Figure - Formato de la PDU AAL5 CPCS.

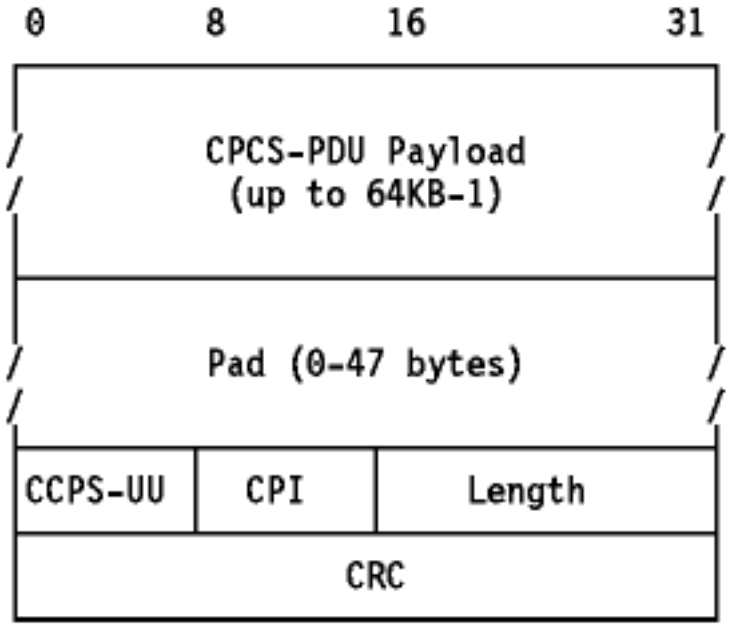


Figure: Formato de la PDU AAL5 CPCS

CPCS-PDU Payload
: Mostrado en Figura - Formato del Payload de CPCS para PDUs de IP.

Pad
: Se utiliza como relleno para que el tamaño de CDCS se ajuste a las cedas ATM.

CPCS-UU
: El campo CPCS-UU ("User-to-User identification") se usa para transmitir de forma transparente información de usuario a usuario. Este campo no tiene utilidad en la encapsulación y se le puede dar cualquier valor.

CPI
: El campo CPI ("Common Part Indicator") alinea la cola del CPCS a 64 bits.

Length
: El campo Length indica la longitud, en bytes, del campo Payload. El valor máximo es 65535(64KB - 1)

CRC
: El campo CRC protege todo el CPCS exceptuándose a sí mismo.

El formato del campo Payload para las PDUs enrutadas se muestra en Figura - Formato del Payload de CPCS para PDUs de IP.

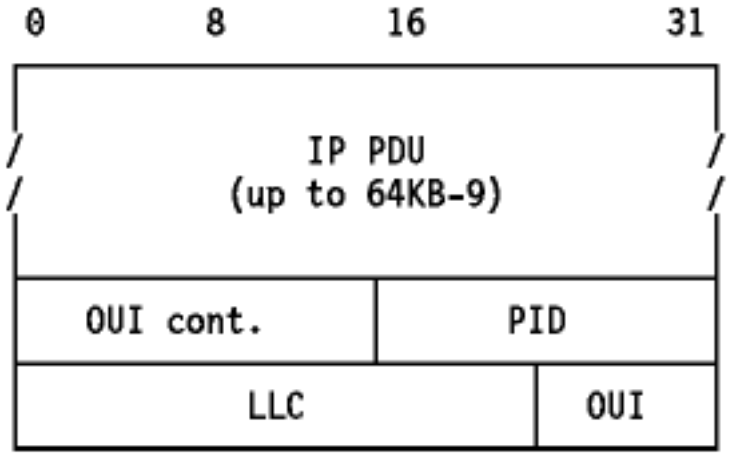


Figura: Formato del Payload de CPCS para PDUs de IP

IP PDU
: Dat IP normal comenzando con la cabecera IP.

LLC
: Cabecera LLC de 3 bytes con el formato DSAP-SSAP-Ctrl. Para datos IP se pone a 0xAA-AA-03 para indicar la presencia de una cabecera SNAP. El campo Ctrl tiene siempre el valor 0x03 especificando una PDU de tipo "Unnumbered Information Command".

OUI
: El campo de 3 bytes OUI ("Organizationally Unique Identifier") identifica una organización que administra el significado

del PID(explicado a continuación). Para especificar el tipo EtherType en el PID el OUI tiene que ponerse a 0x00-00-00.

PID

El PID("Protocol Identifier") de 2 bytes el tipo de protocolo de la PDU que le sucede. Para datagramas IP, el PID es 0x08-00.

El tamaño por defecto de la MTU para miembros IP de la red ATM se discute en el RFC 1626 y se fija a 9180 bytes. La cabecera LLC/SNAP es de 8 bytes; por consiguiente, el tamaño por defecto de la PDU ATM AAL5 es de 9188 bytes. Se puede cambiar el tamaño de la MTU, pero debe hacerse por igual para todos los miembros del LIS. El RFC 1755 recomienda que todas las implementaciones soporten tamaños de MTU de hasta 64KB, inclusive.

No hay forma de mapear direcciones IP de broadcast o de multicast a ATM. Sin embargo, no hay restricciones para transmitir o recibir datagramas especificando cualquiera de las cuatro dirección de broadcast estándar de IP descritas en el RFC 1122. Los miembros, al recibir un broadcast IP para su LIS, deben procesar el paquete como si fuera para ellos.

2.13.3 Emulación LAN con ATM

Otra forma de proporcionar un medio de migrar a una red ATM nativa es la emulación LAN por ATM. Aún está en fase de diseño, de manos de los grupo de trabajo del Forum de ATM. Remitirse a IP clásico sobre ATM para conocer el punto de vista del IETF.

No existe consenso en lo que respecta a la implementación de una LAN virtual sobre ATM, aunque si hay algunos puntos de común acuerdo acerca de las distintas propuestas hechas al Forum de ATM.

 Tabla de contenidos  TCP/IP y OSI

2.14 TCP/IP y OSI

Figura- TCP/IP y OSI muestra un intento de establecer una correspondencia entre las diferentes capas de las arquitecturas de TCP/IP y OSI, pero hay que ser consciente de las diferencias básicas explicadas más abajo.

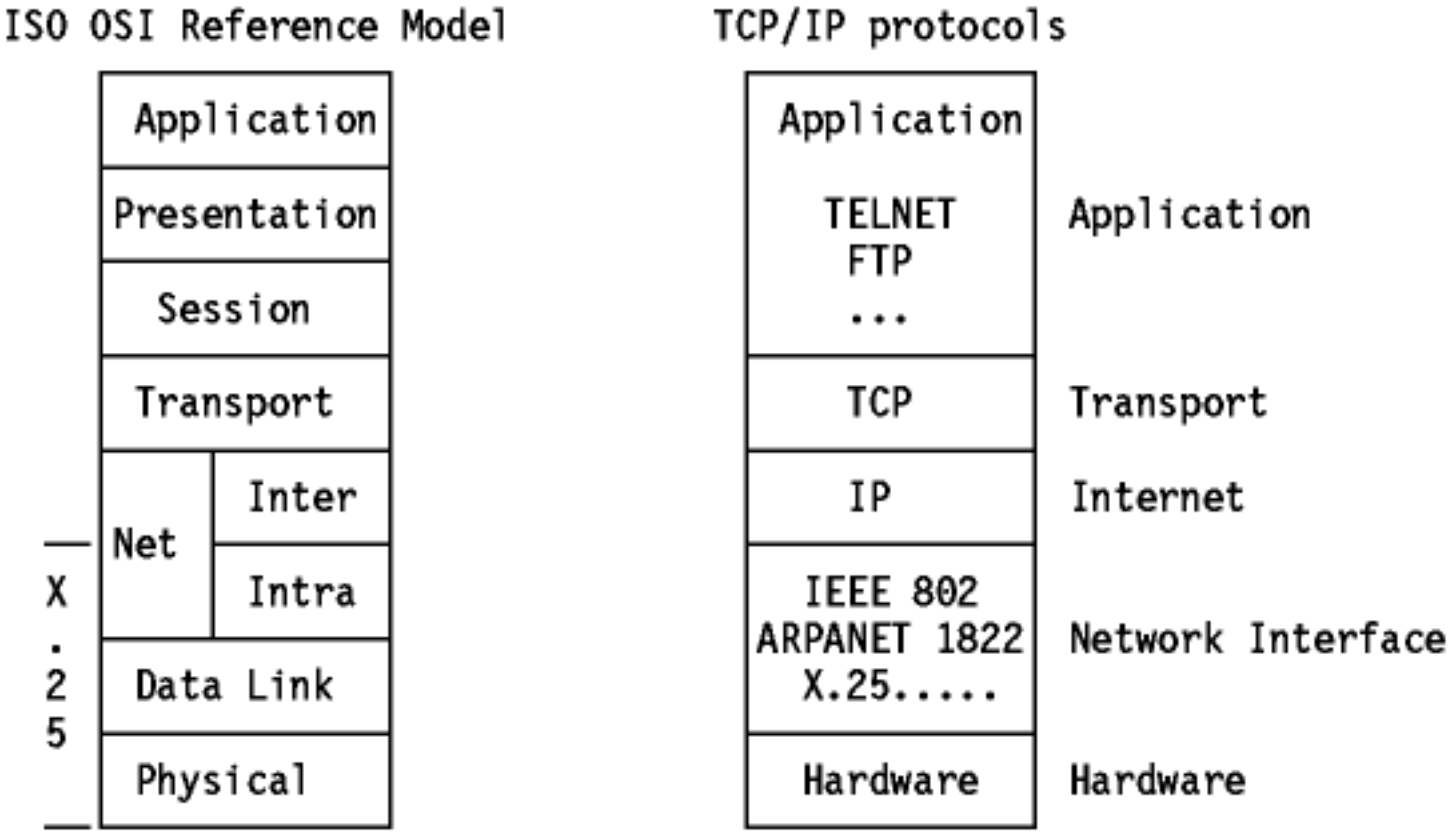


Figura: TCP/IP y OSI - Correspondencia funcional de las capas.

2.14.1 Diferencias

El modelo de internet sólo puede equipararse funcionalmente al modelo OSI de ISO, ya que existen diferencias básicas tales como:

- En la pila de protocos de internet, una capa representa un encapsulamiento de una función.

  La perspectiva de ISO, por otro lado, trata a las capas como grupos funcionales bastante reducidos, intentando forzar la modularidad al requerir capas adicionales para funciones adicionales.

  En los protocolos TCP/IP, un protocolo dado puede ser usado por otros protocolos en la misma capa, mientras que en el modelo OSI se definiría dos capas en las mismas circunstancias. Ejemplos de estas "dependencias horizontales" son FTP, que usa la misma representación común que TELNET sobre la capa de aplicación, o ICMP, que usa IP para el envío de datagramas en el nivel de red.

  A nivel práctico, lo que estamos discutiendo aquí es la diferencia entre un estándar "de jure", OSI, y uno "de facto", TCP/IP. El objetivo en el mundo de TCP/IP consiste en establecer de común acuerdo un protocolo estándar que pueda funcionar en una diversidad de redes heterogéneas; siempre se le ha dado mayor importancia al estándar en sí que a su implementación.

- Eficiencia y viabilidad. Las normas de OSI tienden a ser prescriptivas(por ejemplo, la capa "N" debe atravesar todas las capas "por debajo" de ella), mientras que los protocolos TCP/IP tienden a ser descriptivos, y dejan un máximo de libertad

a los implementadores. Una de las ventajas del enfoque de TCP/IP es que cada implementación concreta puede explotar características dependientes del sistema, de lo que suele derivarse una mayor eficiencia(menos ciclos de CPU, mayor productividad para las mismas funciones), al mismo tiempo que se asegura la interoperabilidad con otras aplicaciones.

Otra forma de ver esto es que la mayoría de ls protocolos de internet se han desarrollado primero(codificados y testeados) antes de ser descritos en un RFC(habitualmente por parte del implementador) lo que muestra claramente su viabilidad.

2.14.2 El mundo de Internet y OSI

El DoD("Department of Defense"), organismo subvencionador de la investigación ARPANET original, hizo una consideración acerca de OSI en enero del '88; basada en el GOSIP("Government OSI Profile") del 22 de abril de 1987.

A continuación mostramos un extracto del documento, que está publicado en el *RFC 1039 - Una consideración de DoD acerca de OSI*: "...Se pretende adoptar los protocolos OSI como un **co-estándar** de hecho con los protocolo de DoD cuando GOSIP sea aprobado oficialmente como un estándar federal de procesamiento de información("Federal Information Processing Standard"). Dos años después, los protocolos OSI se convertirían en la pila de protocolos interoperables predominante; sin embargo, se proporcionaría la capacidad para interoperar con protocolos DoD a efectos de las expectativas de vida de los sistemas que soportasen protocolos DoD..."

El mundo de internet ha realizado numerosos estudios sobre posibles transiciones y la coexistencia de los protocolos. La siguiente lista es serie de RFCs emitidos con este propósito:

- *RFC 983 - Servicios de transporte ISO en la cima de TCP.*
- *RFC 1006 - Servicios de transporte ISO en la cima de TCP - Versión 3.*
- *RFC 1039 - Una consideración de DoD acerca de OSI.*
- *RFC 1069 - Indicaciones para el uso de direcciones IP de Internet en el Protocolo de Red No Orientado a Conexión de ISO.*
- *RFC 1085 - Servicios de presentación de ISO en la cima de TCP/IP.*
- *RFC 1086 - Puente ISO-TP0 entre TCP and X.25.*
- *RFC 1090 - SMTP en X.25.*
- *RFC 1161 - SNMP enr OSI.*
- *RFC 1195 - Uso de OSI IS-IS para el encaminamiento en TCP/IP y en entornos duales.*
- *RFC 1238 - CLNS MIB para uso con el ISO 8473("Connectionless Network Protocol") e ISO 9542("End System to Intermediate System").*
- *RFC 1240 - Servicios de transporte no orientados a conexión de OSI sobre UDP: versión 1.*
- *RFC 1308 - Introducción práctica a servicios de directorios("Directory Services")usando el protocolo X.500*.
- *RFC 1309 - Descripción técnica de los servicios de directorios usando el protocolo X.500*.
- *RFC 1327 - Mapeado entre X.400(1998)/ISO 10021 y el RFC 822*.
- *RFC 1328 - X.400 1988 a 1984 "downgrading"*.
- *RFC 1330 - ESCC X.500/X.400 "Task Force". Recomendaciones para la fase 1 de la instalación de los servicios OSI de directorios (X.500) y de manejo de mensajes(X.400) en la comunidad ESnet*.
- *RFC 1430 - Un plan estratégico para instalar un servicio de directorios de Internet X.500*.
- *RFC 1487 - X.500 "Lightweight Directory Access Protocol"*.
- *RFC 1488 - La representación con cadenas de las sintaxis de atributos estándar de X.500("String Representation of Standard Attribute Syntaxes")*.
- *RFC 1491 - Descripción de usos avanzados de X.500*.
- *RFC 1562 - Nombrando indicaciones para el servicio de directorios AARNet X.500*.
- *RFC 1567 - X.500 monitorizando MIB*.
- *RFC 1608 - Representando información IP en X.500*.
- *RFC 1609 - Esquematizando redes en X.500*.
- *RFC 1617 - Nombrando y estructurando directrices para el X.500 "Directory Pilots"*.
- *RFC 1629 - Directrices para OSI NSAP "Allocation" en Internet*.
- *RFC 1632 - Un catálogo revisado de implementaciones disponibles de X.500*.
- *RFC 1684 - Introducción a los servicios de páginas amarillas basados en X.500*.
- *RFC 1706 - DNS NSAP Registros de recursos*.
- *RFC 1729 - Usando el protocolo de rcuperación de información Z39.50 en Internet*.
- *RFC 1781 - Usando el directorio OSI para el servicio "Achieve User Friendly Naming"*.

2.14.3 Consideraciones acerca de la coexistencia de TCP/IP y OSI

Si se tiene como meta la coexistencia de TCP/IP y OSI, con vistas a la hegemonía eventual de OSI, básicamente hay cinco opciones, divididas en dos categorías generales: basadas en protocolos y basadas en servicios.

Los enfoques basados en protocolos incluyen pilas duales y pasarelas para los niveles de aplicación y de transporte. Los basados en servicios incluyen puentes en el nivel de transporte y túneles de red.

2.14.3.1 Pilas duales - Un enfoque sencillo

La forma más simple de integrar TCP/IP y OSI e poner ambos protocolos en cada máquina de la red.

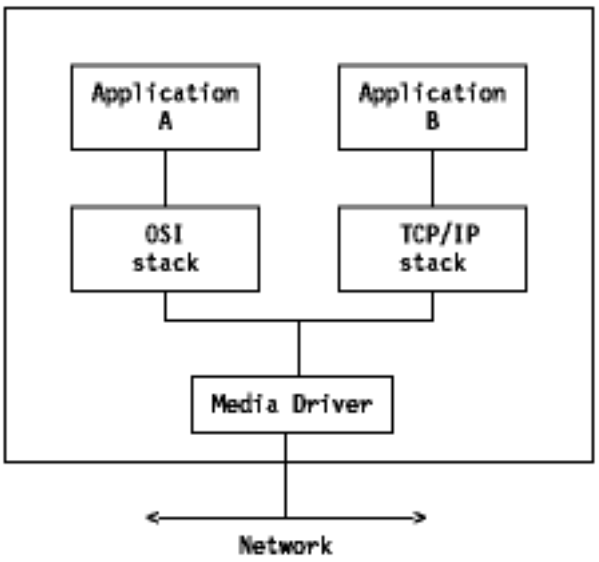


Figura: Pilas duales

Aunque este una perspectiva relativamente directa, supone una gran desventaja: dos redes usarán el mismo conjunto de líneas físicas, pero los dos conjuntos de protocolos no podrán interoperar. Los usuarios están forzados a elegir uno de ellos. Esta desventaja de tener dos redes separadas puede ser una ventaja. Los usuarios que deseen usar TCP/IP podrán hacerlo; los que prefieran OSI podrán usar OSI. Otra ventaja es que una red TCP/IP ya existente no será perturbada. No obstante, las pilas duales consumen mucha memoria, y tienen que instalarse en cada máquina en la red TCP/IP - OSI.

2.14.3.2 Pasarelas en el nivel de aplicación - El enfoque de DoD

Este enfoque elimina una gran desventaja de las pilas duales(la falta de interoperabilidad de aplicaciones) Las pasarelas del nivel de aplicación convierten las PDU("Protocol Data Units") del nivel de aplicación de una pila al de la otra. Estas pasarelas permiten comunicar cualquier sistema TCP/IP con cualquier sistema OSI. No es necesario elegir entre protocolos, ya que el protocolo de aplicación funcionará en ambas pilas.either stack.

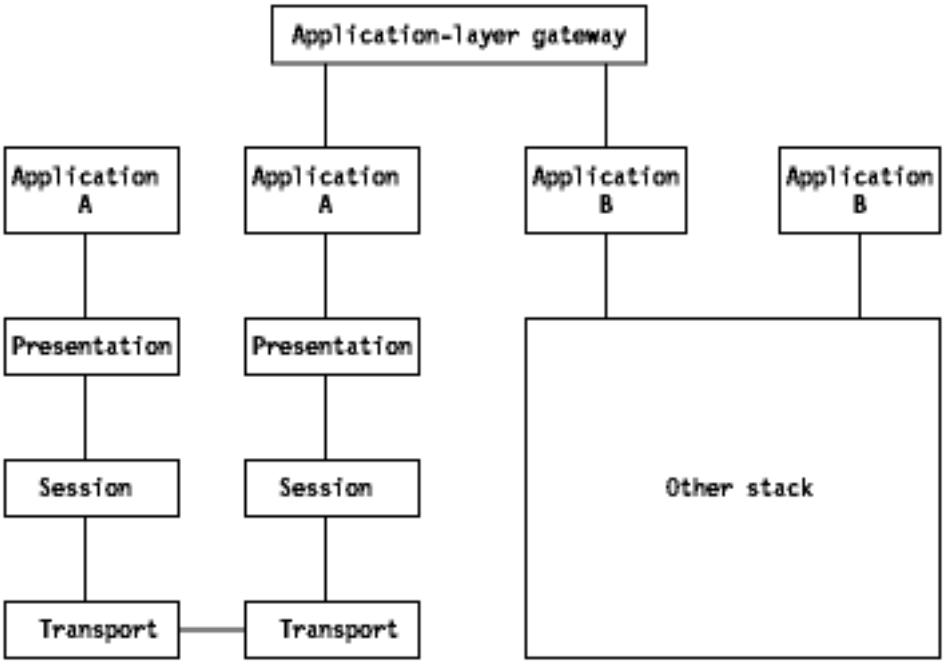


Figura: Nodo pasarela del nivel de aplicación

Otra ventaja de estas pasarelas es que no tienes que añadir o modificar nada en los sistemas en los extremos. Esto se debe a que la pasarela(que incluye las dos pilas de protocolos) actúa de intermediaria para el sistema y maneja todas las conversiones de protocolo. Pero se suele perder funcionalidad en la conversión de un protocolo de aplicación a otro debido a que la correspondencia o mapeo entre los dos protocolos no es perfecta, principalmente cuando se va de aplicaciones de OSI a aplicaciones de TCP/IP. Esto se debe a que las aplicaciones OSI tiene mayor exuberancia de funcionalidades. Otra desventaja de las pasarelas es que producen cuellos de botella que dan lugar a una degradación del rendimiento si la pasarela no es lo bastante potente.

2.14.3.3 Pasarelas del nivel de transporte - El enfoque equivocado

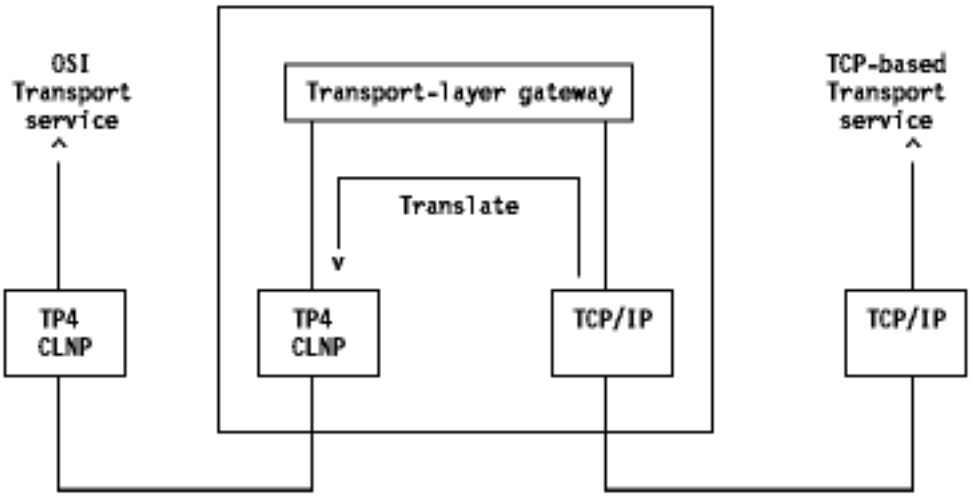


Figura: Nodo pasarela del nivel de transporte

CLNP significa *"ConnectionLess Network Protocol"* (ver figura superior).

Si se dispone de un entorno con el protocolo de transporte TCP en un extremo y del protocolo de transporte de OSI TP4 en otro, hace falta un mecanismo de software que traduzca dinámicamente los paquetes TCP a paquetes TP4. Esta aproximación se considera errónea porque ninguna aplicación soporta TCP en un extremo y TP4 en otro, y porque el cambio de direcciones en cualquier lado de la pasarela, que implica una pérdida en los servicios de directorios.

Los tres enfoques examinados hasta el momento se concentran en la conversión de protocolos. Sin embargo, es posible ignorar virtualmente el protocolo en sí y concentrarse en emular los servicios. Aquí es donde entran en juego los puentes del servicio de transporte y los túneles de red.

2.14.3.4 Puentes del servicio de transporte - El enfoque de ISODE

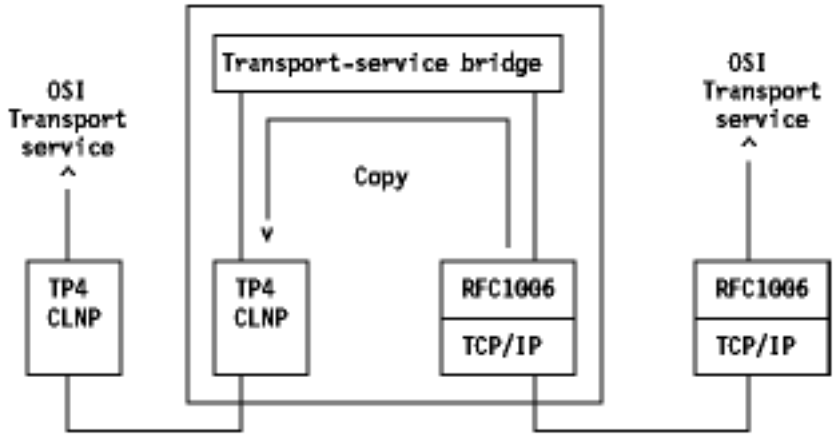


Figura: Nodo puente del servicio de transporte

ISODE significa *"International Standards Organization Development Environment"*. Se trata de una colección disponible al público de librerías de rutinas y programas que implementan los servicios de las capas superiores de OSI.

En un ejemplo de TCP a TP4, un puente del sevicio de transporte haría que el servicio de TCP parezca un servicio TP4. Con los puentes del servicio de transporte, es posible ejectuar aplicaciones OSI sobre redes TCP/IP. Una ventaja de esto es que sólo se necesita un protocolo de aplicación - el de OSI. Sólo las capas subyacentes necesitan cambiar entre los dos entornos. Un puente del servicio de transporte es esencialmente un "router" que copia PDUs, en vez de traducirlas. El RFC 1006 define la forma de generar servicios de transporte de OSI sobre TCP. La principal desventaja de esto este enfoque es que no hay campo checksum origen - destino. En el ejemplo de entorno TCP a TP4, se tiene una checksum de TCP en el lado correspondiente al origen del puente y un checksum en el lado de destino. Como las pasarelas, los puentes del sevicio de transporte tienen un sólo punto débil. Es posible usar estos puentes no sólo para implementaciones TCP/IP a OSI sino también para trabajos de integración de OSI a OSI. Por ejemplo, OSi incluye diferentes protocolos de transporte, como TP= para WANs y TP4 para LANs. Los puentes del servicio de transporte son candidatos viables para conectar esos sistemas OSI con diferentes esquemas de transporte.

2.14.3.5 Túneles de red - El enfoque complejo

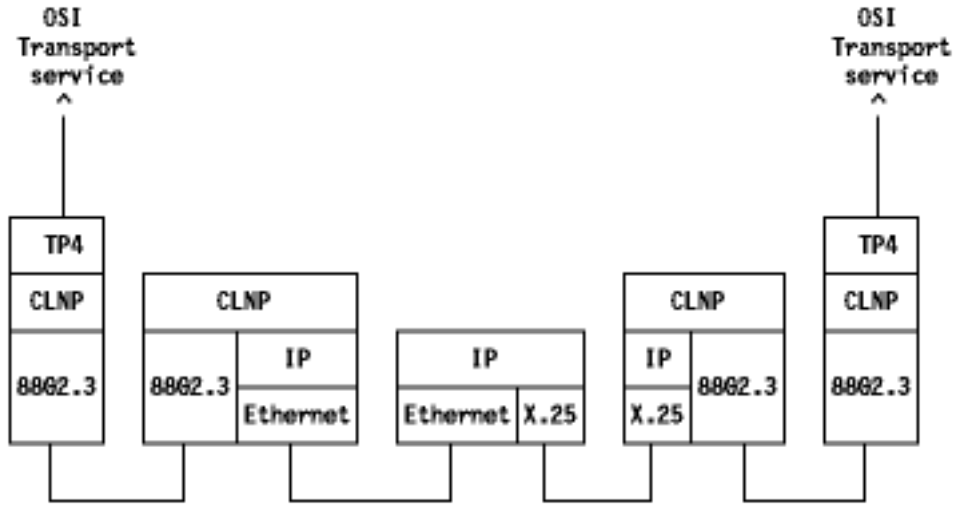


Figura: Túneles de red

Esta aproximación elimina el único punto débil y proporciona checksums origen - destino. Los túneles de red están un nivel por debajo del punto de vista de los puentes del servicio de transporte. En vez de emular el servicio de transporte, usan una emulación del servicio del nivel de paquetes. Los túneles de red operan al nivel de red y no al de transporte. Encapsulan los paquetes OSI CLNP en paquetes IP y los hacen circular sobre redes IP. Los túneles de red son esencialmente "routers" CLNP.

Los túneles de red permiten tener checksums origen - destino, un mayor grado de transparencia, pero requieren redes basadas en OSI CLNP y son difíciles de implementar.

 Tabla de contenidos  IP: la próxima generación(IPng)

2.16 IP: La próxima generación(IPng)

Internet ha crecido rápidamente en los últimos años, y en diciembre de 1994 albergaba más de 32000 redes que conectaban más de 3.8 millones de ordenadores en más de 90 países. Como un dirección de 32 bits proporciona unos 4 billones de posibles direcciones, da la impresión de el esquema de direccionamiento IP es más que suficiente para todos los hosts de Internet puesto que parece que aún es posible un incremento en mil veces la ocupación del espacio de direcciones. Desafortunadamente, no es este el caso, por una serie de razones, entre las que están las siguientes:

- La dirección IP se divide en un número de red y en una parte local que se administra por separado. Aunque el espacio de direcciones dentro de una red puede estar poco ocupado, en lo que respecta el espacio efectivo, si se usa un número de red, todas las direcciones de esa red están ocupadas.
- El espacio de direcciones para redes se estructura en las clases A, B y C, de distintos tamaños, y deben considerarse por separado los espacios de cada una.
- El modelo de direccionamiento IP requiere que todas las redes IP tengan números unívocos, estén o no conectadas a Internet.
- El crecimiento del uso de TCP/IP en nuevas áreas podría resultar en una rápida explosión del número de direcciones IP requeridas. Por ejemplo, la difusión de TCP/IP para interconectar terminales de venta o receptores de televisión por cable incrementaría enormemente el número de hosts IP
- El modelo actual de direccionamiento IP con una sola dirección para cada host(que no sea "router") podría cambiar en el futuro(ver *RFC 1681 -- Cómo tener muchas direcciones por host* para conocer posibles formas de realizar este cambio).

Estos factores implican que el espacio de direcciones está mucho más constreñido de lo que se supuso al principio. Este problema se denomina *Agotamiento de las direcciones IP*. Ya se están aplicando métodos para solucionarlo(ver El problema del agotamiento de las direcciones IP) pero al final, el espacio actual acabará por agotarse. El IETF("Internet Engineering Task Force") tiene un grupo de trabajo en *ALE("Address Lifetime Expectations")* con el fin de proporcionar estimaciones de cuándo el problema será intratable, y las estimaciones actuales fijan el agotamiento del espacio de direcciones entre el 2005 y el 2011. Antes de que esto ocurra, hará falta un sustituto para la versión actual de IP. Como además es posible que cambien las tendencias en e uso de IP, puede que esta sustitución sea necesaria para el año 2000. Es lo que se conoce como *IP: La próxima generación(IPng)* . La versión actual de IP es *IPv4*. La responsabilidad de la forma final de IPng la tiene el *IPngD("IPng Directorate")*. Hay otros grupos de trabajo en IP: *IPNGREQ("IPng Requirements")*, *TACIT("Transition and Co-existence including Testing")* y un grupo para candidato a IPng. Estos grupos son todos temporales y se espera que se disgreguen o que se fusionen con otros grupos de trabajo en otras áreas en las que el proceso de definición de IPng ya esté completo.

2.16.1 Requerimientos de IPng

En julio de 1994, en un mitin del IETF en Toronto, el IPngAD("IPng Area Directors") del IETF presentó el *RFC 1752 - Las recomendaciones para el protocolo IPng*. Fue aprobado por el IETF en noviembre de 1994 y se convirtió en *estándar propuesto*.

Estos sucesos fueron la culminación de mucho trabajo y discusión que implicó a muchas partes interesadas.

El IPngD publicó el *RFC 1550 - Solicitudes para IPng* pidiendo requerimientos para IPng. Los requerimientos importantes se resumen aquí:

- Un espacio de direcciones mucho más grande: al menos 10exp9 redes, preferiblemente 10exp12; y al menos 10exp12 hosts, preferiblemente 10exp15. Esto permitiría una drástica reducción en el uso de las direcciones IP y al mismo tiempo dejaría el espacio de direcciones poco poblado permitiendo que las direcciones IPng estén más estructuradas que en IPv4.
- IPng debería permitir la encapsulación de sus propios paquetes o de los de otros protocolos.
- IPng debería añadir clases de servicio para distinguir los tipos de datos transmitidos, tales como tráfico isócrono como audio y vídeo en tiempo real.
- IPng debe proporcionar direccionamiento multicast de forma que esté más completamente integrado con el resto de la pila que IPv4.
- IPng debe proporcionar autentificación y encriptación.
- IPng debería preservar las virtudes de IPv4: robustez, independencia de las características de la red física, alto rendimiento, topología flexible, extensibilidad, servicio de datagramas, direccionamiento unívoco a nivel global, protocolo de control integrado y estándares de libre distribución.
- La implementación debe suponer una transición sencilla.
- IPng debe coexistir con IPv4.

2.16.2 Candidatos a IPng

Hay tres propuestas principales para IPng:

2.16.2.1 CATNIP("Common Architecture for the Internet")

CATNIP es un desarrollo de un viejo protocolo(TP/IX) que integra IPv4, Novell IPX y OSI CLNP("Connectionless Networking Protocol") y proporciona una infraestructura común. Se acerca en diseño a CLNP y enfatiza la facilidad de interoperabilidad con las implementaciones existentes de los tres. El paquete CATNIP contiene toda la información requerida por los tres protocolos en un formato comprimido usando una cabecera de 16 o más bytes. CATNIP usa una dirección de longitud variable. Las direcciones IPv4 existentes se mapean a direcciones de 7 bytes de las cuales sólo los últimos 4 bytes son la dirección IPv4. Los host IPv4 existentes deberían limitarse a interoperar de esta forma con los hosts

CATNIP.

CATNIP se describe en el *RFC 1707 - CATNIP: Una arquitectura común para Internet*.

2.16.2.2 TUBA("TCP and UDP with Bigger Addresses")

TUBA también se basa en CLNP; en pocas palabras, sustituye a IPv4 en la pila TCP/IP. Enfatiza las redes multiprotocolo. La transición entre IPv4 a IPng se hace usando una estrategia de pila dual. La pila de protocolo tiene dos capas de red independientes y cuando se intenta comunicar con otro host, un host de pila dual consulta al DNS sobre la dirección IP y el *NSAP("Network Service Access Point")* que es el equivalente CLNP. Si el DNS devuelve tanto la dirección IP como el NSAP, el host se comunica usando CLNP como protocolo de red.

TUBA se describe en el *RFC 1347 - TUBA("TCP and UDP with Bigger Addresses"), una simple propuesta para el direccionamiento y el encaminamiento en Internet*. Ver también el *RFC 1526 - Asignación de identificadores de sistema para hosts TUBA/CLNP* y el *RFC 1561 - Uso de ISO CLNP en entornos TUBA*.

2.16.2.3 SIPP("Simple Internet Protocol Plus")

SIPP es una combinación del trabajo de tres grupos de trabajo anteriores del IETF que dedicados al desarrollo de un IPng.

IPAE("IP Address Encapsulation")
: IPAE determina que las extensiones de IPv4 lleven direcciones más largas, y como debe realizarse la transición de una versión a otra.

SIP("Simple Internet Protocol")
: SIP es una sustitución de IPv4 con una cabecera IP simplificada y direccionamiento de 64 bits, unido a IPAE usando la cabecera SIP y los mecanismos de transición IPAE.

PIP(""P" Internet Protocol")
: Pip fue una nueva marca para un protocolo de Internet, diseñado con una amplia variedad de características avanzadas y usando direccionamiento de longitud variable. PIP se unió con SIP cuando quedó patente que las mejores características de PIP se podían usar con el esquema de direccionamiento de 64 bits de SIP y los mecanismos de transición de IPAE.

SIPP es un desarrollo evolutivo de IPv4. Enfatiza la eficiencia de operatividad en una gran variedad de tipos de red y la facilidad de interoperabilidad. Además del direccionamiento de 64 bits, incluye el concepto de direcciones extendidas usando una opción de encaminamiento: la longitud de la dirección efectiva puede ser cualquier múltiplo de 64.

SIPP se describe en el *RFC 1710 - SIPP("Simple Internet Protocol Plus White Paper")*.

2.16.3 Versión 6 de IP(IPv6)

El IPngD("IPng Directorate") concluyó que las tres propuestas eran insuficientes para(CATNIP, TUBA y SIPP) para afrontar la lista aceptada de requirimientos, pero que SIPP, según como se define en el RFC 1710, era el que más se acercaba. Tras algunos cambios en el protocolo original, por ejemplo el uso de direcciones de 128 bits en vez de 64,, se decidió que SIPP era una base adecuada para IPng y que se le añadirían las características de otras propuestas para completar los restantes requerimientos. La solución propuesta se llamó *IP Versión 6(IPv6)*.

Su definición sigue aún en progreso, y la información presentada aquí se basa en borradores. (6)

****** Aviso ******

Toda la información de la sección IP Versión 6 (IPv6) está sujeta a cambio y no se puede usar como referencia.

Se espera que la definición final de IPv6 se publique como series de estándares en RFCs.

****** Terminología ****** IPv6 usa el término *paquete* más que *datagrama*, pero el significado es el mismo, aunque los formatos sean distintos.

IPv6 introduce un nuevo término, *nodo*, para un sistema que ejecuta IPv6, es decir, un host o un "router". Un host IPv6 es un nodo que no envía paquetes IPv6 que no estén dirigidos explícitamente a él. Un "router" es, como en IPv4, un nodo que envía paquetes no dirigidos explícitamente a él.

2.16.3.1 El formato de la cabecera IPv6

IPv6 incrementa la longitud de la cabecera IP de 20 a 40 bytes. La cabecera IPv6 contiene dos direcciones de 16 bytes(fuente y destino) precedidas de 8 bytes de control como se muestra en Figura - Cabecera IPv6. La cabecera IPv4 (ver Figura - Formato del datagrama IP) tiene dos direcciones de 4-byte precedidas de 12 bytes de control y seguidos posiblemente opciones. La reducción de la información de control y la

eliminación de opciones de la cabecera tienen como fin optimizar el procesamiento del paquete. Los campos de uso poco frecuente que se han eliminado de la cabecera se han pasado a extensiones de cabecera opcionales.

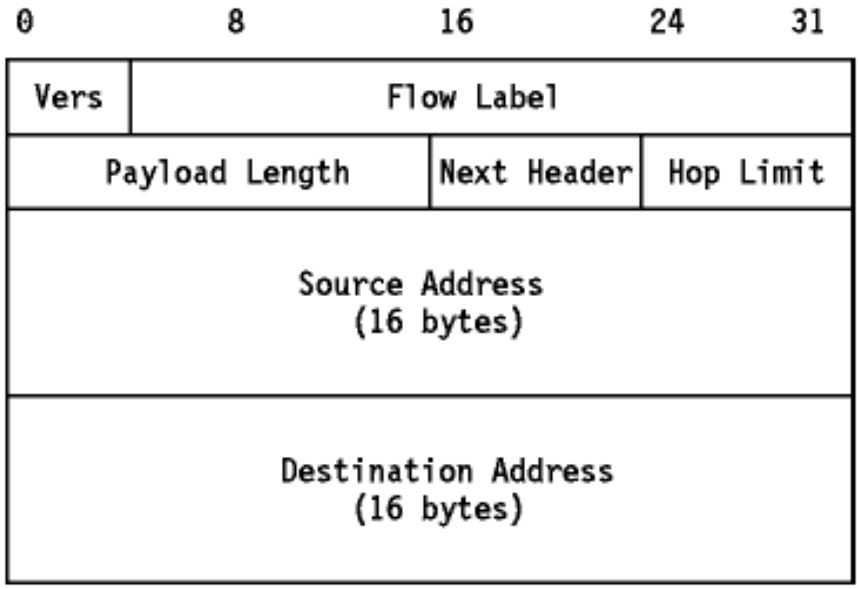


Figura: Cabecera IPv6

Vers
: Número de versión de 4 bits: 6.

Flow Label
: Campo de 28 bits. Ver Etiquetas de flujo más abajo.

Payload Length
: La longitud del paquete en bytes(exluyendo la cabecera) codificada como un entero sin signo de 16 bits. Si la longitud es mayor de 64Kb, el campo vale 0 y la cabecera opcional da la verdadera longitud.

Next Header
: Indica el tipo de cabecera que sigue inmediatamente esta cabecera. Es el mismo que el número de protocolo usado en IPv4(ver la lista en IP("Internet Protocol")). El campo "next header" se usa demás para indicar la presencia de cabeceras de extensión, que proporcionan los mecanismos para añadir información opcional al paquete. Los siguientes valores son importantes además de los mencionados para IPv4.

41
: IPv6 Header

43
: IPv6 Routing Header

44
: IPv6 Fragment Header

51
: IPv6 Authentication Header

?
: IPv6 End-to-End Options Header

?
: IPv6 ICMP Packet

Los valores, excepto los dos últimos(que no estaban decididos en el momento de redactar el documento) figuran en el *STD 2 - Números asignados de Internet*, aunque la edición actual del STD 2(RFC 1700) menciona como protocolo tanto a SIP como a SIPP. Como se indicó antes, IPv6 es un desarrollo de estos dos protocolos.

Los distintos tipos de EH("extension header") se discuten brevemente más abajo.

Hop Limit
: Es es el TTL de IPv4 pero ahora se mide en saltos y no en segundos. Se cambió por dos razones:

- ❏ IP envía normalmente los datagramas a más de un salto por segundo y el campo TTL se decrementa siempre en cada salto, por lo que a efectos prácticos se mide en saltos y no en segundos.
- ❏ Muchas

implementaciones de IP no causan la expiración de los datagramas en base al tiempo transcurrido.

Source Address
: Una dirección de 128 bits. Las direcciones IPv6 se discuten en Direcciones IPv6.

Destination Address
: Una dirección de 128 bits. Las direcciones IPv6 se discuten en Direcciones IPv6.

Una comparación entre los formatos de las cabeceras de IPv4 y IPv6 muestra que los campos de IPv4 no tienen equivalente en IPv6.

Type of Service
: Los problemas del TOS/("type of service") de IPv6 se manejan usando el concepto de *flujo*, descrito en Etiquetas de flujo.

Identification, Fragmentation Flags y Fragment Offset
: Los paquetes fragmentados tienen una extensión en vez de información de fragmentación en la cabecera IPv6. Esto reduce el tamaño de la cabecera básica IPv6. Debido a que los protocolos de alto nivel, TCP en particular, tienden a evitar la fragmentación de datagramas, se reduce el overhead para la cabecera Ipv6 en los casos normales. Como se indicó arriba, IPv6 no fragmenta los paquetes en ruta, sólo en el destino.

Header Checksum
: Como los protocolos de transporte implementan checksums, y como IPv6 incluye una cabecera opcional que se puede usar para asegurar la integridad, IPv6 *no* proporciona monitorización del checksum de sus paquetes.
Tanto TCP como UDP incluyen una pseudocabecera IP en los checksums que utilizan, por lo que en estos casos, la cabecera IP en IPv4 se chequea dos veces.
TCP y UDP, y cualquier otro protocolo que use el mismo mecanismo, seguirán usando una pseudocabecera en Ipv6, obviamente, el formato de la pseudocabecera IPv6 será distinto de IPv4. ICMP e IGMP y cualquier otro protocolo que no emplee la pseudocabecera en IPv4 tendrán que usarla en Ipv6 para su checksums.

Options
: Todos los valores opcionales asociados con los paquetes IPv6 están contenidos en EHs("extension headers") asegurando que la cabecera básica tiene siempre el mismo tamaño.

2.16.3.2 Tamaños de los paquetes

Se espera que todos los nodos IPv6 determinen dinámicamente la PMTU("Path MTU") de cada enlace(como se describe en el *RFC 1191 - Cáculo de PMTUs*) y los nodos fuente sólo enviarán paquetes que no excedan el tamaño del PMTU. Por ello, los "routers" IPv6 no tendrán que fragmentar paquetes en mitad de rutas con más de un salto y permitirán hacer un uso mucho más eficiente de las rutas. En la actualidad está propuesto que cada enlace soporte una MTU de 576 bytes, pero este valor, como el resto de las especificaciones de IPv6, está sujeto a cambios.

2.16.3.3 EH("Extension Headers")

Los EHs se sitúan entre la cabecera IPv6 y los datos destinados al protocolo de la capa superior. Se cuentan como parte de la carga efectiva de la trama. Cada cabecera tiene un campo de 8 bits, *Next Header*, que identifica el tipo de la cabecera siguiente. Todas las extensiones conocidas tienen este campo como el primer byte de la cabecera. La longitud de cada cabecera, que siempre es un múltiplo de 8 bytes, se codifica más tarde en el formato específico a ese tipo de cabecera. Existe un número limitado de EHs en IPv6, cualquiera de las cuales pueden aparecer sólo una vez en el paquete. Los nodos que originan paquetes deben tener las EHs en un orden específico, no así los que los reciben. Abajo se discuten brevemente los diversos tipo de EHs. Cuando el campo Next Header contiene un valor que no se corresponde con un EH significa el fin de las cabeceras IPv6 y el comienzo de los datos.

IPv6 permite la encapsulación de IPv6 en IPv6("tunneling"). Esto se hace con una valor 41 para el campo Next Header. El paquete IPv6 encapsulado puede tener sus propias EHs. Los "routers" no deberían añadir EHs a los paquetes. Sino encapsularlos en paquetes propio(fragmentados, si hace falta) ya que el tamaño de cada paquete se ha calculado en el nodo originador del mismo para que se ajuste al PMTU.

A excepción de la cabecera Hop by Hop(que debe seguir a la cabecera IP), los EHs sólo se procesan al final del trayecto.

IPv6 usa un formato común llamado *TLV("Type-Length-Value")* para campos de longitud variable hallados en las opciones Hop-by-Hop y End-to-End . Cada opción tiene una cabecera de 2 bytes seguida de los datos de la opción.

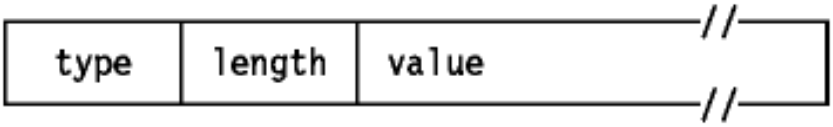
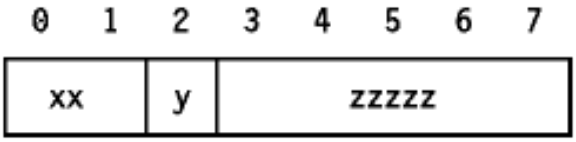


Figura: Formato TLV de IPv6

Type
: El tipo de la opción. Todos tienen un formato común:



xx
: Un número de 2 bits que indica cómo un nodo IPv6 debe tratar una opción que no reconoce.

0
: saltar la opción y continuar

1
: desechar el paquete

2
: desechar el paquete e informar al emisor con un mensaje ICMP Unrecognized Type

3
: desechar el paquete e informar al emisor con un mensaje ICMP Unrecognized Type a menos que la dirección de destino sea de multicast

y
: Este bit tiene un significado especial sólo para la cabecera Hop-by-Hop. Si está activo, indica que el valor de la opción puede cambiar en ruta y por lo tanto debe excluirse de cualquier test de integridad efectuado sobre el paquete. Debido a que sólo los "routers" intermedios examinan estas cabeceras, sólo las opciones hop-by-hop se pueden cambiar en ruta.

zzzzz
: El resto de los bits define la opción.

Length
: Longitud del campo con el valor de la opción en bytes.

Value
: El valor de la opción. Depende del tipo.

Para optimizar el rendimiento de las implementaciones de IPv6, las opciones individuales se alinean de forma que lo valores de varios bytes están situados dentro de sus límites naturales. En muchos casos, esto dará lugar a que estas cabeceras sean mayores de lo necesario, pero debería permitir a los nodos procesar los datagramas con mayor rapidez. Para conseguir esta alineación, todas las implementaciones de IPv6 deben reconocer dos opciones de relleno("padding"):

Pad1
: Un byte X'00' usado para rellenar un sólo byte. Las secuencias de relleno mayores se deberían efectuar con la opción PadN.

PadN
: Una opción en el formato TLV descrito arriba. Su valor es X'01'. El correspondiente campo de longitud("length byte") indica el número de bytes de relleno después de los dos requeridos como mínimo.

Cabecera Hop-by-Hop

Una cabecera Hop-by-Hop contiene opciones que cada nodo que atraviesa el paquete debe examinar, además del nodo de destino. Debe seguir a la cabecera IPv6, y se identifica con el valor 0 en el campo Next Header de la cabecera IPv6. Este valor no es un número de protocolo sino un caso especial para distinguir este tipo único de EH y el valor 0 permanece reservado en el STD 2.

Inicialmente, no están definidas opciones Hop-by-Hop aparte de las de relleno.

RH("Routing Header" o Cabecera de encaminamiento)

Se identifica con el valor 43 en el campo Next Header precedente. Su primer byte es su campo Next Header y el segundo es el Routing Type. El único tipo definido inicialmente es el LSR("Loose Source Routing"), del mismo modo que en IPv4.

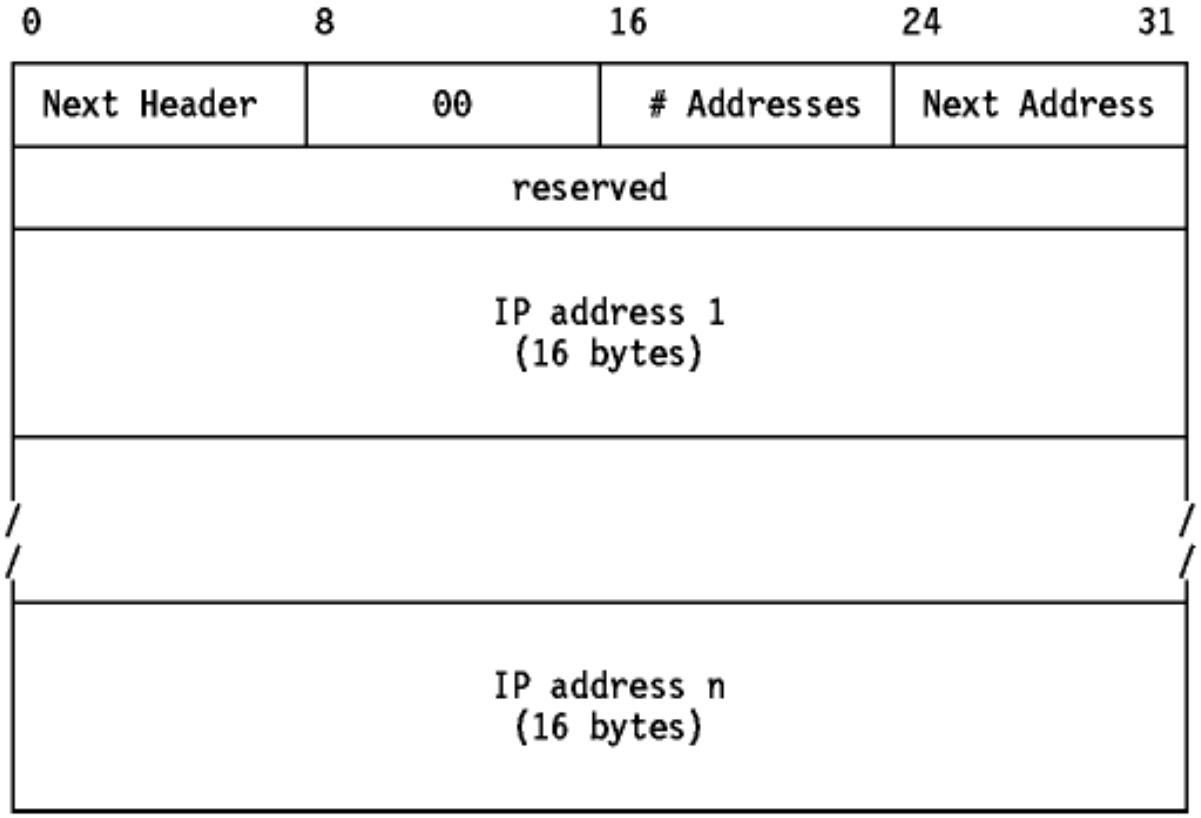


Figura: cabecera LSR de IPv6

Next Header
: El tipo de la siguiente cabecera.

00
: Indica LSR.

#Addresses
: Indica el número de entradas, es un entero sin signo de 8 bits. Compárese con IPv4, que emplea el byte de longitud para calcular el número de entradas del campo de opciones("option field").

Next Address
: Índice a la siguiente dirección, es un entero sin signo de 8 bits que se procesa como entero(el originador lo inicializa a 0).

reserved
: Inicializado a cero para la transmisión, ignorado en la recepción. Asegura que la cabecera tiene una longitud múltiplo de 16 bytes. No asegura que las direcciones estén alineadas a 16 bytes. IPv6 no tienen en cuenta la alineación de campos mayores de 8 bytes.

Address n
: Una serie de direcciones IPv6 de 16 bytes que contienen la ruta especificada por la fuente.

FH("Fragment Header" o cabecera de fragmentación)

Se identifica con 44 en el Next Header precedente.

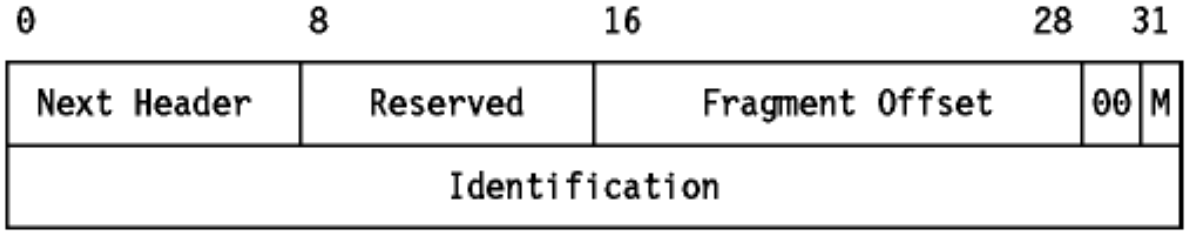


Figura: FH de IPv6

Next Header
: El tipo de la siguiente cabecera.

reserved
Inicializado a cero para la transmisión, ignorado en la recepción.

Fragment Offset
: Un entero sin signo de 13 bits que indica el desplazamiento de la carga efectiva en relación al comienzo de la carga efectiva original sin fragmentar en unidades de 8 bytes.

00
Inicializado a cero para la transmisión, ignorado en la recepción.

M

Flag "More"(más). Si está activo indica que quedan más fragmentos.

Identification
: Usado para identificar paquetes que son fragmentos del mismo datagrama. Muy similar al campo Identifier de IPv4, pero es el doble de largo.

AH("Authentication Header" o cabecera de autentificación)

Se identifica con 51 en el Next Header precedente.

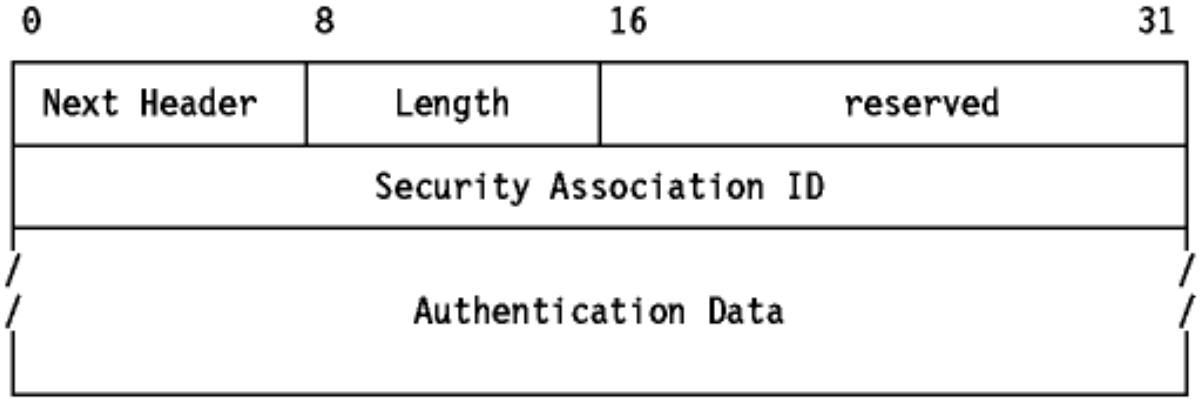


Figura: FH de IPv6

Next Header
: El tipo de la siguiente cabecera.

Length
: Longitud de este campo medida en unidades de 8-byte.

reserved
Inicializado a cero para la transmisión, ignorado en la recepción.

Security Association ID
: Usado para identificar al receptor(con la dirección IP de destino).

Authentication Data
: Dependiente del algoritmo de autentificación. Todos los nodos IPv6 han de soportar un algoritmo mínimo.

Cabecera End-to-End

Tiene el mismo formato que la cabecera Hop-by-Hop, pero sólo lo examina el nodo de destino. Como sigue al RH, es independiente de cualquier opción de encaminamiento. De nuevo, sólo se especificamente inicialmente las opciones de relleno. El valor para el campo Next Header precedente no se ha definido aún.

2.16.3.4 Direcciones IPv6

IPv6 proporciona direcciones de 128 bits de longitud. A diferencia de IPv4, que tiene una organización basada estrictamente en las clases de direcciones, las direcciones IPv6 no están estructuradas así. Están diseñadas para ser usadas con CIDR("Classless InterDomain Routing"; ver CIDR("Classless Inter-Domain Routing")). El espacio de direcciones IPv6 es lo bastante grande como para acompasar un amplio rango de espacios de direcciones existentes y propuestos. La estrategia seguida, análoga a CIDR, consiste en utilizar parte de la dirección, por ejemplo el primer byte, para indicar el tipo de dirección. Estos tipos incluirían un mapeado del espacio IPv4 a IPv6, OSI NSAPs, Novell IPX, etc. Es más, la cabecera IPv6 permite encapsular información de direccionamiento arbitraria en cada paquete, lo que se podría utilizar para extender el mecanismo IPv6 a sistemas hipotéticos que no se pudieran mapear a IP.

2.16.3.5 Etiquetas de flujo("Flow Labels")

IPv6 introduce el concepto de *flow* como una serie de paquetes relacionado desde una fuente a un destino que requiere un tipo particular de manipulación de mano de los "routers" implicados, por ejemplo para un servicio en tiempo real. La naturaleza de esa manipulación se puede implementar con opciones añadidas a los datagramas o con un protocolo separado.

Cada paquete IPv6 contiene una etiquete de flujo que es un campo de 28 bits:

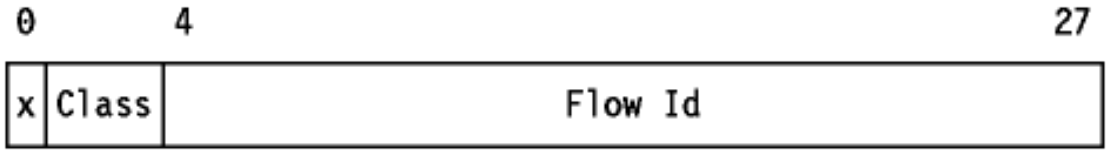


Figura: Etiqueta de flujo de IPv6

x
: Un bit de flag que indica si el tráfico tiene control de flujo . Si está activo, no hay control de flujo(no hay realimentación desde los

receptores), si no lo está, existe control de flujo(por ejemplo, el paquete contiene segmentos TCP).

Class
: Un número de 3 bits indicando el tipo de tráfico. Aunque el protocolo usado para controlar el flujo puede redefinir los valores, los siguientes valores están recomendados para el tráfico con control de flujo:

0
: tráfico sin caracterizar

1
: tráfico "filler"

2
: transferencia de datos no atendida, como E-mail

3
: reservado

4
: transferencia de bloques de datos atendida, como FTP

5
: reservado

6
: tráfico interactivo, como TELNET

7
: tráfico de control de Internet, como protocolos de encaminamiento

Para tráfico sin control de flujo, el valor de Class se usa como prioridad cuando hay algún problema. Cuanto más bajo sea, menos ha de preocuparse el emisor de que el paquete llegue a su destino.

Flow ID
: Un número único pseudoaleatorio de 24 bits, que el nodo fuente asigna a un flujo. El valor cero se usa para el tráfico que no tiene asociado un flujo. La aleatoriedad es necesaria para permitir a los "routers" emplear una secuencia de bits del flujo como clave de dispersión(hash).

2.16.3.6 SIT("Simple Internet Transition")

Las técnicas para convertir Internet de IPv4 a IPv6 se denominan en conjunto SIT(*"Simple Internet Transition")*. SIT enfatiza la facilidad del proceso desde el punto de vista del administrador y del usuario. Las características de compatibilidad aseguran la protección de inversiones para los usuarios actuales de IPv4, las de interoperabilidad que la transición sea gradual y no impacte sobre la funcionalidad de Internet. La transición emplea las siguientes técnicas:

- Implementaciones de pila dual para los host y "routers" que deben interoperar entre IPv4 e IPv6.
- Direcciones IPv4 embebidas en direcciones IPv6, a los hosts IPv6 se les asignan direcciones compatibles con IPv4 y las direcciones IPv4 se mapean a IPv6.
- Un mecanismo de "tunneling" de IPv6 sobre IPv4. Sólo para cuando la implementación de IPv6 esté bien avanzada. Permite la creación de nodos sólo IPv6, que deben existir en redes IPv6 completamente funcionales.
- Traducción de cabeceras IPv4/IPv6 en los "routers" situados entre redes IPv4 e IPv6.

Las técnicas son adaptables a otros protocolos, notablemente, CLNP e IPX que tienen semánticas similares a nivel de red y con esquemas de direccionamiento fácilmente mapeables a IPv6.

El modelo de transición vislumbra la migración de diferentes organizaciones de forma independiente y en dos fases. La primera es una transición a una infraestructura dual IPv6/IPv4. La segunda, que no es obligatoria, es a una infraestructura sólo IPv6. Sólo se completa cuando no sea necesaria la interoperabilidad con IPv4.

La primera etapa es la más fácil de las dos, ya que todos los nodos soportan IPv4. La segunda requiere más esfuerzo, particularmente en la planificación e instalación de los "routers" que realizarán la traducción de cabeceras adecuada para que los nodos IPv6 interoperen con nodos IPv4.

Instalación de nodos IPv6/IPv4

Implica reemplazar el software de nodos sólo IPv4 por software IPv6/IPv4. Esto debería formar parte de los ciclos de distribución habituales, y los nodos IPv4 continuarían funcionando en modo compatible con IPv4.

Conceptualmente, el modelo de pila dual establece una duplicidad en los protocolos de la capa de red. Sin embargo, los cambios relacionados se necesitan obviamente en todas los protocolos de transporte para operar usando ambas pilas, y posiblemente en las aplicaciones, si se pretende que estas exploten las posibilidades de IPv6, como por ejemplo la mayor longitud de las direcciones.

Direccionamiento IPv4/IPv6

Notación: las direcciones IPv6 se representan como una secuencia de 4 dígitos hexadecimales separados por comas. La secuencia 0000 se contrae como 0. Las direcciones IPv6 que se han de mapear a IPv4 se representan mejor como un prefijo IPv6 de 96 bits separado por comas y seguido de una dirección IPv4 en formato decimal separado por puntos, por ejemplo 0:0:0:0:0:ffff:9.180.214.114

Se definen tres tipos de direcciones IPv6:

Compatibles con IPv4
: Una dirección indicando un nodo IPv6 con una dirección que se puede mapear unívocamente al espacio IPv4. Tienen el prefijo IP 0:0:0:0:0:ffff. Por ejemplo, 0:0:0:0:0:ffff:9.180.214.114

Mapeadas a IPv4
: Una dirección IPv6 que indica un nodo sólo IPv4. Tienen el prefijo IP 0:0:0:0:0:0. Por ejemplo, 0:0:0:0:0:0:9.180.214.114. Es importante darse cuenta de que las direcciones compatibles con IPv4 y las mapeadas a IPv4 utilizan el mismo espacio de direcciones. El prefijo sólo indica si el nodo soporta o no IPv6.

Sólo IPv6
: Una dirección IPv6 que indica un nodo que soporta IPv6 donde los 32 bits inferiores no contienen necesariamente una dirección IPv4. Los 96 bits de orden superior son distintos de 0:0:0:0:0:ffff o 0:0:0:0:0:0.

Para el DNS se define un nuevo tipo de registro RR, AAAA, que indica una dirección IPv6. Los registros que el DNS encuentre en un nodo dependen del protocolo que se esté ejecutando.

- Los nodos sólo IPv4 sólo tienen registros A que contienen direcciones IPv4 en el DNS. Esto simplifica la administración DNS y significa que el comportamiento por defecto es el compatible: los nodos IPv6 interpretan los contenidos del DNS pre-IPv6 como nodos IPv4. Un nodo IPv6 puede obtener la dirección IPv6 de cualquier nodo IPv4 en el DNS, prefijándolo con los 96 bits siguientes: 0:0:0:0:0:0.
- Los nodos IPv6 que pueden interoperar con nodos sólo IPv4 tienen registros AAAA que contienen direcciones IPv6 compatibles con IPv4 y registros A con las direcciones IPv4 equivalentes.
- Los nodos IPv6 que no pueden interoperar con los que son sólo IPv4 tienen registros AAAA con las direcciones sólo IPv6.

Debido a que los nodos IPv6/IPv4 toman decisiones acerca de qué protocolos usar basadas en el tipo de dirección IPv6 del destino, la incorporación de registros AAAA al DNS es un prerrequisito para usar el DNS con IPv6. Esto no implica que los servidores de nombres tengan que usar pilas IPv6, sólo que soporten un tipo adicional de registro.

Resumen de interoperabilidad

El que dos nodos puedan interoperar depende de sus capacidades y de sus direcciones:

- Un nodo IPv6 con una dirección compatible con IPv4 puede interoperar directamente con otros nodos.
- Un nodo sólo IPv6 con una dirección compatible con IPv4 puede interoperar con el resto de los nodos. Sin embargo, requiere un "router" para traducir las cabeceras IPv6 a cabeceras IPv4 y viceversa con el fin de interoperar con nodos IPv4.
- Un nodo IPv6 con una dirección sólo IPv6 no puede interoperar con nodos IPv4.
- Un nodo sólo IPv4 puede interoperar directamente con nodos IPv6/IPv4 con direcciones compatibles con IPv4.
- Un nodo sólo IPv4 puede interoperar con nodos sólo IPv6 que tengan direcciones compatibles con IPv4. Sin embargo, requiere que un "router" traduzca las cabeceras IPv4 a IPv6 y vice versa.
- Un nodo sólo IPv4 no puede interoperar con nodos IPv6 que tengan direcciones sólo IPv6.

Modelo topológico

La topología de encaminamiento de Internet se puede dividir en áreas de tal forma que cada área caiga en al menos dentro de uno de estos dos tipos:

IPv4-complete
: Cada subred del área tiene conectado al menos un "router" IPv4.

IPv6-complete
: Cada subred del área tiene conectado al menos un "router" IPv6.

Un área puede ser de ambos tipos, aunque el modelo se simplifica mucho si se cada área se trata sólo como uno de los dos tipos. Se emplean las siguientes reglas para la instalación de nodos que no usen pila dual:

- Los nodos sólo IPv4 no se pueden instalar en áreas de tipo IPv6-complete.
- Los nodos sólo IPv6 no se pueden instalar en áreas de tipo IPv6-complete.
- Todos los "routers" que conecten áreas IPv4-complete con áreas IPv6-complete deben traducir las cabeceras IPv4 destinadas al área IPv6-complete a cabeceras IPv6, vice versa.

Nota: Que un área sea de tipo IPv4-complete no significa que no se pueda usar encaminamiento IPv6 en ella, sólo que se usa encaminamiento IPv4 en su totalidad. Lo inverso es cierto en áreas tipo IPv6-complete.

Como se indicó arriba, la instalación de "routers" traductores de cabeceras es parte de la segunda fase de la transición. Por lo tanto, es improbable que las áreas IPv6-complete aparezcan inmediatamente en las organizaciones. Posiblemente, los controladores para la introducción de áreas IPv6-complete sean requerimientos para nuevos servicios que requieren IPv6, o para el agotamiento del espacio IPv4. La importancia de estos dos factores variará según la organización. Por ejemplo, las organizaciones comerciales con grandes redes IPv4 internas seguramente no se verán afectadas por el problema del agotamiento de las direcciones IPv4 a menos que lo experimenten dentro de sus propias redes.

"Tunneling" de IPv6-sobre-IPv4

El "tunneling" de paquetes IPv6 sobre IPv4 es muy simple; el paquete IPv6 se encapsula en un datagrama IPv4(que puede estar fragmentado).

Hay dos clases de "tunneling" de paquetes IPv6 sobre redes IPv4: automático y configurado.

Automático

Como el nombre indica, se realiza siempre que sea necesario. La decisión la toma un host IPv6/IPv4 que tiene un paquete que enviar a través de un área de tipo IPv4-complete, y sigue las siguientes reglas:

- Si el destino es una dirección mapeada a IPv4, envía el paquete usando IPv4 ya que el receptor no soporta IPv6. En otro caso:
- Si el destino se halla en la misma subred, lo envía usando IPv6, ya que el receptor soportará IPv6.
- Si el destino no se halla en la misma subred pero en ella hay al menos un "router" por defecto que soporta IPv6, o existe una ruta configurada para un "router" de este tipo, el paquete se le envía a este "router" mediante IPv6. En otro caso:
- Si la dirección es compatible con IPv4, el paquete se envía usando"tunneling" automático. En otro caso:
- El destino es un nodo sólo IPv6 conectado a un área tipo IPv4-complete que no sea IPv6-complete y es por tanto inaccesible.

Nota: La dirección IP debe ser compatible con IPv4 para que se puede usar "tunneling". Las direcciones sólo IPv6 no son susceptibles de ser usadas ya que no se las puede direccionar con IPv4. Los paquetes de nodos IPv6/IPv4 a direcciones mapeadas a IPv4 no se "tunelizan" ya que se refieren a nodos sólo IPv4.

Las reglas anteriores enfatizan el uso de un "router" IPv6 con preferencia al "tunneling" por tres razones:

- Hay menos overhead ya que no existe encapsulación con cabeceras IPv4.
- Sólo están disponibles características de IPv6.
- La topología de encaminamiento IPv6 se usará desde su instalación, preferentemente a la de IPv4.

Un nodo no necesita conocer si está conectado a una área IPv6-complete o IPv4-complete: siempre que sea posible usará un "router" IPv6, y si no lo es usará "tunneling" (y en este caso deducirá que está conectado a un área IPv4-complete).

El "tunneling" automático puede ser host-a-host, o "router"-a-host. Un host fuente enviará un paquete IPv6 a un "router" IPv6 si es posible, pero puede que el "router" no pueda hacer lo propio, y en ese caso tendrá que hacer "tunneling". Debido a la preferencia de IPv6 sobre el "tunneling", los túneles serán siempre tan cortos como sea posible. Sin embargo, el túnel siempre se extiende por la ruta hasta el host de destino: como IPv6 usa el paradigma de encaminamiento de "saltos", un host no puede determinar si el paquete acabará por llegar a un área IPv6-complete antes de llegar al host de destino. Para evitar túneles que se extiendan por toda la ruta hasta el receptor, se emplea "tunneling" configurado. Hay una excepción a esta regla: como se describe en Traducción de cabeceras, todos los túneles terminan en "routers" que realizan traducción de cabeceras.

El mecanismo usado para el "tunneling" automático es muy simple.

- El datagrama IPv4 que encapsula emplea los 32 bits de orden inferior de las direcciones IPv6 fuente y destino para crear sus equivalentes IPv4 y fija el número de protocolo a 41(IPv6).
- La interfaz de red del receptor identifica los paquetes que van llegando como paquetes IPv4 y los pasa a la parte IPv4 de la capa de red dual.
- La capa IPv4 recibe el datagrama del modo normal, reensambla los fragmentos si hace falta, nota que el protocolo es el 41, elimina la cabecera IPv4 y pasa el paquete IPv6 original a la parte IPv6 de la capa de red.
- El código IPv6 procesa el paquete original de la forma habitual. Como la dirección IPv6 de destino del paquete es la dirección IPv6 del nodo(y es compatible con IPv4) el paquete ha llegado a su destino. IPv6 procesa cualquier EH y pasa el resto de la carga efectiva del paquete al siguiente protocolo listado en la cabecera IPv6.

Con una excepción ,descrita en Traducción de cabeceras, los "routers" IPv6/IPv4 intermedios nunca examinan los contenidos del paquete IPv6 encapsulado. El datagrama IPv4 se trata como si fuera un datagrama IPv4 normal y corriente.

"Tunneling" configurado

Se utiliza para realizar "tunneling" host-router o router-router de IPv6-sobre-IPv4. El host emisor o el "router" retransmisor se configuran de tal forma que la ruta, además de contar con el siguiente salto, tiene una dirección de "fin de túnel"(que es siempre compatible con IPv4 ya que se debe tratar de un host IPv6/IPv4 accesible desde un área IP-complete). El proceso de encapsulación es el mismo que en modo automático exceptuando que la dirección IPv4 de destino no se calcula de los 32 bits inferiores de la dirección IPv6, sino de los 32 bits inferiores del fin de túnel. Cuando el "router" del fin del túnel recibe el datagrama, lo procesa como lo haría un nodo que estuviera al final de un túnel automático. Cuando el paquete IPv6 original para a la capa IPv6 del "router", este reconoce que no es el destinatario y lo retransmite como lo haría con cualquier paquete IPv6.

Puede ocurrir, por supuesto, que tras emerger de un túnel, el paquete entre nuevo en otro.

Traducción de cabeceras

La traducción de cabeceras la requieren los nodos sólo IPv6 para interoperar con nodos sólo IPv4. Es una parte opcional de SIT. La llevan a cabo los "routers" IPv6/IPv4 dituados en las fronteras entre áreas IPv4-complete e IPv6-complete. El tráfico que cruza la frontera se clasifica de formas. Primero, el tráfico es:

IPv4
: Trafico de un área IPv4-complete que entra en un área IPv6-complete o

IPv6

Trafico de un área IPv6-complete que entra en un área IPv4-complete.

Segundo, cada uno de estos tipos se puede describir como:

Terminal
Dirigido a un nodo dentro del área o
Tránsito
Dirigido a un nodo fuera del área

Los "routers" traductores tienen que seleccionar la forma adecuada de direcciones IP, además de mapear correctamente las direcciones a traducir:

- Las direcciones IPv4 se obtienen tomando los 32 bits de orden inferior de la dirección IP. Si la fuente o el destino son sólo IPv6, la cabecera es intraducible.
- Las direcciones fuente IPv6 se crean añadiendo el prefijo de 96 bits 0:0:0:0:0:0 a la dirección IPv4.
- Las direcciones destino IPv6 se crean añadiendo el prefijo de 96 bits 0:0:0:0:0:ffff a la dirección IPv4 para generar una dirección IPv6 compatible con IPv4 para el tráfico terminal o el prefijo de 96 bits 0:0:0:0:0:0 para una dirección IPv6 mapeada a IPv4 para el tráfico de tránsito. En consecuencia, los traductores de cabeceras deben conocer los límites de su área. (7)

Hay un caso especial: el tráfico IPv6 "tunelizado". Si los traductores de cabeceras lo tratasen como tráfico IPv4 normal, el resultado sería un paquete IPv6 encapsulado en otro paquete IPv6. Por ello, los traductores han de examinar el número de protocolo de los datagramas IPv6, y si es 41(IPv6), desencapsularán el paquete en vez de traducir la cabecera IPv4. Efectivamente, los traductores de cabeceras siempre terminan los túneles.

Debido a este efecto, no es posible, en general, que un nodo IPv6/IPv4 envíe un paquete a un nodo con una dirección sólo IPv6 usando un túnel configurado a un "router" IPv6/IPv4 en la misma área IPv6-complete que el destino. Si se hiciera, el túnel podrían intersectar un área IPv6-complete de tránsito y el paquete se desencapsularía, terminando el túnel, y el paquete IPv6 no podría cruzar el área IPv4-complete debido a su dirección de destino sólo IPv6.

Para que un nodo IPv6/IPv4 envíe un paquete sólo IPv6 por medio de un "router" que soporte IPv4 en la misma área IPv6-complete que el destino, el paquete debe contener una ruta IPv6 especificada por la fuente consistente en el "router" IPv4 y el destino sólo IPv6. Este paquete tiene una dirección IPv4 compatible hasta que alcance el área IPv6-complete destino, para que pueda ser "tunelizado" seguramente a través de IPv4 con independencia de la topología.

Simetría del modelo SIT

Aunque el modelo topológico de SIT es simétrico, ya que su clasificación de los nodos es sólo IPv6, sólo IPv4, IPv4/IPv6, hay otros aspectos del diseño que no lo son:

- Aunque todas las direcciones IPv4 tienen un equivalente IPv6, lo inverso no es cierto. Los hosts con direcciones sólo IPv6 no pueden interoperar con IPv4.
- Los hosts IPv6/IPv4 usarán IPv6 con preferencia sobre IPv4 de ser posible.
- SIT no define un mecanismo para el "tunneling" de IPv4-sobre-IPv6.

Tabla de contenidos Resumen

Tabla de contenidos Simetría del modelo SIT

2.17 Resumen

El siguiente esquema muestra el modelo por capas de la pila TCP/IP y además indica las APIs disponible para el usuario.

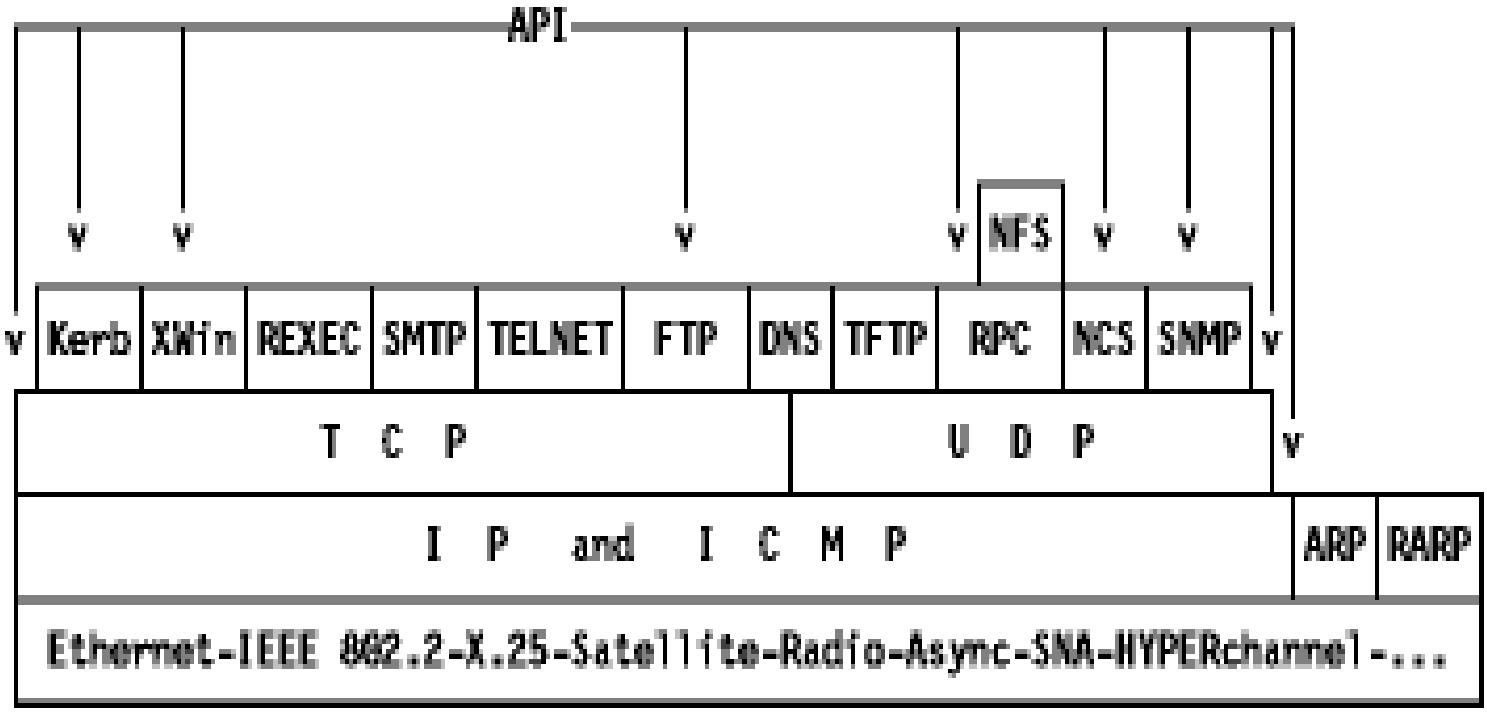


Figura: Modelo por capas de TCP/IP

Nota: RPC usa tanto TCP como UDP. Se ha colocado sobre UDP porque NFS sólo utiliza RPC sobre UDP.

Nota: ARP y RARP se usan sólo en redes de área local.

La API para zócalos IP/TCP/UDP se he mencionado en Puertos y zócalos. Las otras APIs se explicarán en el siguiente capítulo:

- Kerberos (ver Sistema de autenticación y autorización Kerberos)
- X Window (ver Sistema X Window)
- FTP (ver FTP("File Transfer Protocol "))
- RPC (ver RPC("Remote Procedure Call"))
- NCS (ver NCS("Network Computing System"))
- SNMP DPI (ver Gestión de red)
- Interfaz de zócalos CICS (ver Interfaz de zócalos CICS).

Tabla de contenidos Protocolos de encaminamiento

Capítulo 3. Protocolos de encaminamiento

Una de las funciones básicas de IP es su habilidad para conectar distintas redes físicas. Esto es posible gracias a la flexibilidad de IP para utilizar para usar casi cualquier red física, y a su algoritmo de encaminamiento. Al sistema que realiza esta función se le denomina *router*, aunque también se emplea el término *pasarela IP*.

Nota: En otras secciones de este documento se muestra la posición de cada protocolo en el modelo de capas de la pila TCP/IP. El encaminamiento forma parte de la capa de red, pero la función principal de un protocolo de encaminamiento es intercambiar información con otros routers, y en este sentido los protocolos se comportan como si fueran de aplicación. Todos los protocolos escritos aquí emplean tres estrategias para el transporte de datos: usar UDP (por ejemplo RIP, descrito en Routing Information Protocol (RIP)), TCP (ver BGP en Border Gateway Protocol (BGP)) o bien crear su propia capa de transporte sobre IP (ver OSPF en Open Shortest Path First Protocol (OSPF) Version 2). Por lo tanto, no tiene mucho sentido representar la posición de estos protocolos en la pila como se ha hecho con otros.

3.1 Encaminamiento IP básico

La función fundamental de encaminamiento está presente en *todas* las implementaciones de IP:

- Un datagrama IP *entrante* que especifica una "ir IP de destino" distinta de la dirección local del host se trata como un datagrama IP *saliente* normal y corriente.

Este datagrama IP está sujeto al algoritmo de encaminamiento IP del host local, que selecciona el *siguiente salto* del datagrama(el siguiente host al que se enviará). Este nuevo destino puede estar en cualquiera de las redes físicas con las que el host está conectado. Si es una red física diferente de aquella en la que se recibió el datagrama, resulta que el host que hace de intermediario ha *retransmitido* el datagrama de una red física a otra.

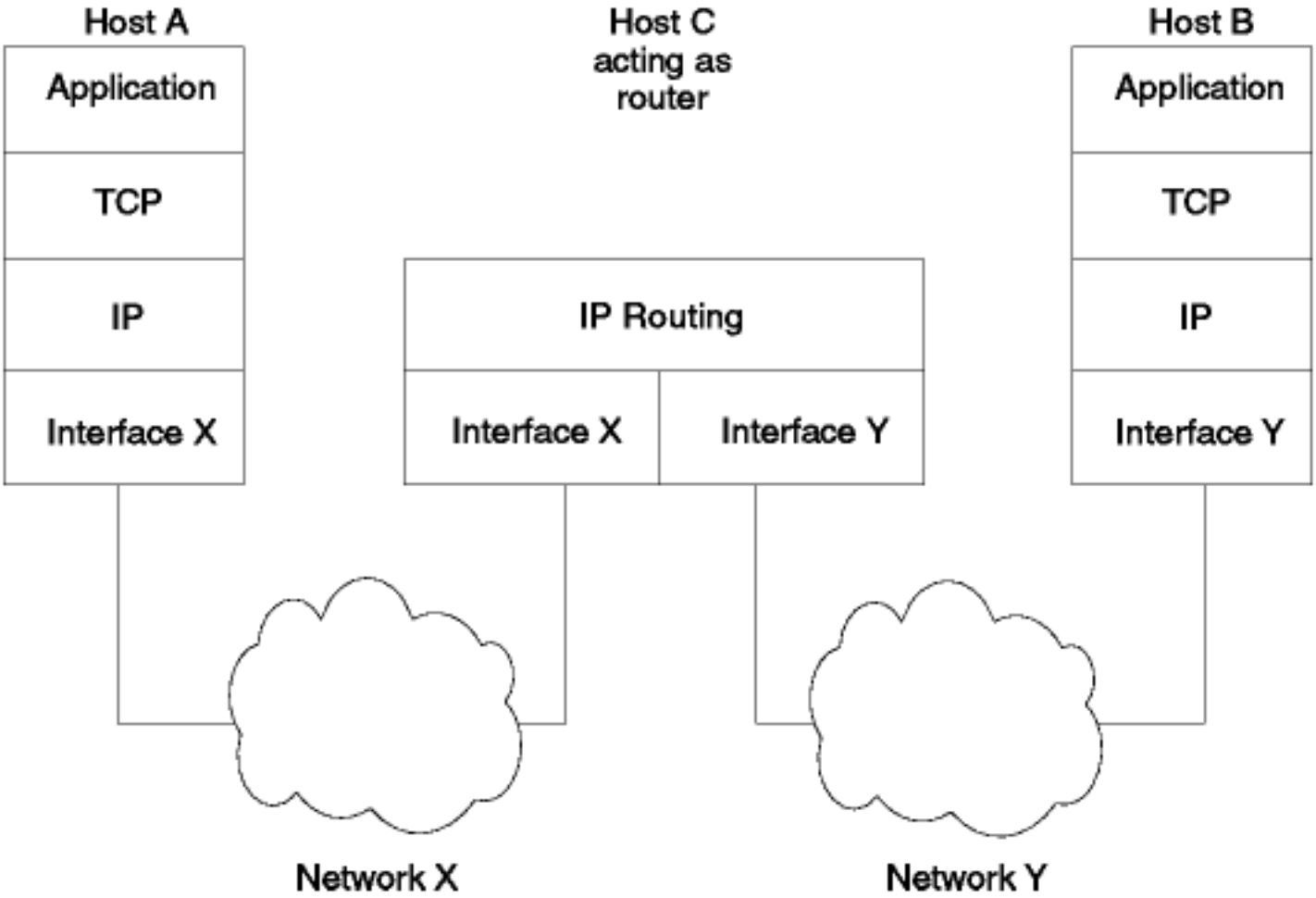


Figura: Funcionamiento del "router" en IP - *Todas* las implementaciones de IP lo pueden hacer.

La tabla de encaminamiento IP normal contiene información acerca de las redes conectadas localmente y de las direcciones IP de otros "routers" localizados en ellas, además de las redes con las que están conectados. Se puede extender con información de las redes IP que se hallan aún más lejos, y tener incluso una ruta por defecto, pero sigue representando una fracción de Internet. Por ello se le llama *"router" con información parcial de encaminamiento*.

A estos "routers" se les aplican algunas consideraciones:

- No conocen todas las redes de Internet.
- Permiten la autonomía de sitios locales para establecer y modificar rutas.
- Una entrada de encaminamiento errónea en uno de los "routers" puede introducir inconsistencias, haciendo por tanto que parte de la red sea inalcanzable.

Deberían implementar algún mecanismo de informe de errores vía ICMP("Internet Control Message Protocol") descrito en

ICMP("Internet Control Message Protocol"). Los siguiente errores deberían poderse enviar al host fuente:

- Destino IP desconocido con un mensaje ICMP *Destination Unreachable*.
- Redirección del tráfico a "routers" más adecuados enviando mensajes ICMP *Redirect*.
- Problemas de congestión(demasiados datagramas entrantes para el espacio disponible en el buffer) con el mensaje ICMP *Source Quench*.
- El campo TTL("Time-to-Live") de un datagrama IP ha llegado a cero. Se comunica con el mensaje ICMP *Time Exceeded*.
- Además, se deberían soportar las siguientes operaciones y mensajes ICMP básicos:
  - Problema de parámetros
  - Máscara de dirección
  - TS("Time stamp")
  - Solicitud/repuesta de información
  - Solicitud respuesta de eco

Hace falta un "router" más inteligente si:

- Ha de conocer las rutas a *todas* las posibles redes IP, como era el caso de las pasarelas del *núcleo* de ARPANET.
- El "router" a de tener tablas de encaminamiento dinámicas, que se actualizan con poca o ninguna intervención manual.
- El "router" ha de anunciar los cambios locales a los otros "routers".

Estas formas más avanzadas de "router" usan protocolos adicionales para comunicarse. Existen cuatro razones básicas para esta multiplicidad de protocolos:

- Usando la terminología de Internet, existe el concepto de *AS("Autonomous System")*, o grupo de redes, que se administra como una unidad.

  Los encaminamientos dentro y fuera de un AS se tratan como cuestiones distintas y existen protocolos diferentes para cada uno.

- Durante unas dos décadas se probaron numerosos protocolos en Internet. Algunos de ellos funcionaron bien, otros tuvieron que ser abandonados.
- La aparición de ASs se distintos tamaños requería diferentes soluciones de encaminamiento. Para ASs de tamaño medio se hizo muy popular un grupo de protocolos basado en DV("Distant Vector"), como por ejemplo RIP. Sin embargo, no funcionaban bien en redes grandes. Los protocolos LS("Link State") como OSPF son mucho más adecuados para estas redes.
- Para intercambiar información de encaminamiento entre ASs se desarrollaron los protocolos de pasarelas fronterizas.

Antes de discutir los diversos protocolos de encaminamiento, repasaremos las arquitecturas de encaminamiento primigenias de Internet, ya que así se comprenderán mejor los papeles que juegan los distintos protocolos. Esta descripción también muestra la diferencia entre encaminamiento *Interior* y *Exterior*. Luego se discutirán los variados protocolos usados para cada uno de estos dos tipo.

3.1.1 "Demonios" de encaminamiento

Los protocolos de encaminamiento suelen implementarse con dos tipos de demonios: (8)

routed("route D")
: Se trata de un demonio de encaminamiento interior básico que se suministra con la mayoría de las implementaciones de TCP/IP. Utiliza el protocolo RIP(ver RIP("Routing Information Protocol")).

gated("gate D")
: Es un demonio más sofisticado para encaminamiento interior y exterior en sistemas basados en UNIX. Puede emplear un número de protocolos adicionales tales como OSPF (ver OSPF ("Open Shortest Path First Protocol"), versión 2) y BGP (ver BGP("Border Gateway Protocol")).

Tabla de contenidos Perspectiva histórica

3.2 Perspectiva histórica

Inicialmente, ARPANET(ver ARPANET) jugó un papel central en el desarrollo de Internet, particularmente en el área del encaminamiento. Aunque fue sustituida en su papel de troncal por NSFNET a finales de los 80', la experiencia ganada de su arquitectura tuvo un efecto directo en el desarrollo posterior del conjunto actual de protocolos de encaminamiento.

3.2.1 La arquitectura de encaminamiento ARPANET

Su esencia reside en el concepto de AS("*Autonomous System"):* una colección de redes controladas por una única autoridad. Cada AS está registrado en el InterNIC y tiene un número de identificación llamado número de AS. Están listado en el *RFC 1166 - Nums de Internet*. El propio núcleo de ARPANET se consideraba un AS.

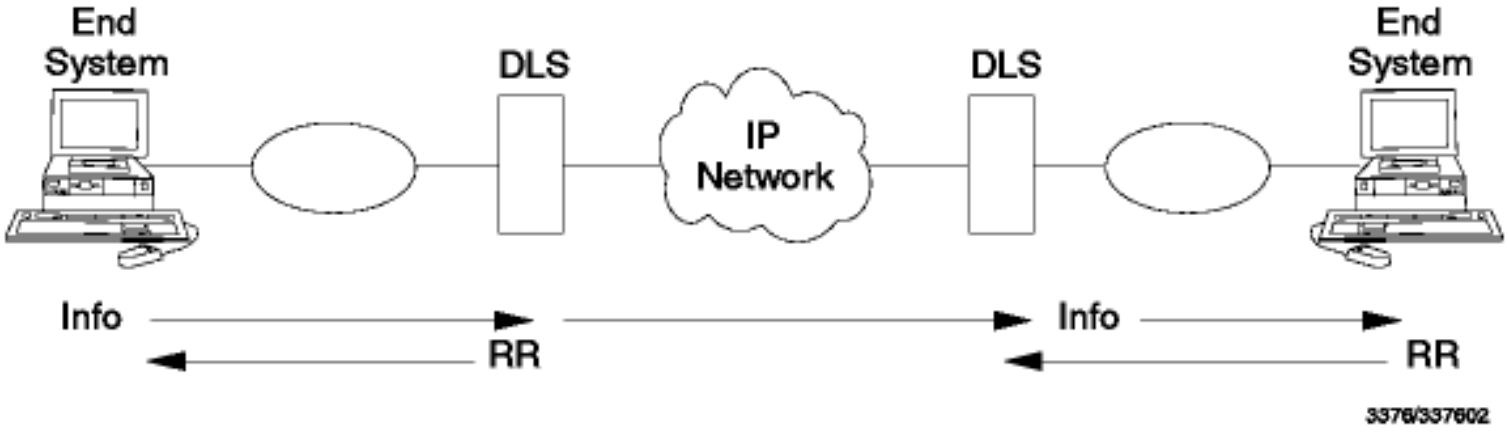


Figura: La troncal ARPANET - Los términos del diagrama se explican a continuación.

Para mantener la nomenclatura, a los "router" entre ASs se les llamará pasarelas. Todo el encaminamiento entre pasarelas se puede categorizar en *intra-AS* (o *Interior*) si las pasarelas pertenecen al mismo AS, o *inter-AS* (o *Exterior*) si pertenecen a distintos ASs. El encaminamiento intra-AS usa un IGP("*Interior Gateway Protocol")* y el inter-AS usa un EGP("*Exterior Gateway Protocol").* La arquitectura ARPANET no especifica que protocolo se debería usa como EGP.

Para más confusión, al prot usado como EGP se le llamó también EGP.

Nota: Para evitar confusión, se usará el término "EGP" para hacer referencia al prot EGP, y "un EGP" para referirse a un protocolo pertenciente al grupo de prottocolos EGP.

3.2.1.1 Pasarelas nucleares y no nucleares

En el sist ARPANET, las pasarelas nucleares que constituían la troncal eran mantenidas por una autoridad central, el INOC("Internet Network Operations Center"). Proporcionaban rutas fiables para todas la posibles redes de Internet, y conectaban ARPANET con el resto de las redes de Internet.

Las pasarelas nucleares(CGx, CGy, etc., en Figura - La troncal ARPANET) tenían que saberlo todo acerca de sus destinos para optimizar el tráfico de ARPANET. Un datagrama viajando de una red local a otra a través de una del núcleo pasaba exactamente por dos pasarelas. La arquitectura de encaminamiento ARPANET especificaba que estas pasarelas se comunicasen con GGP*("Gateway-to-Gateway Protocol")*.

Pasarelas no nucleares (mostradas como G en Figura - La troncal ARPANET) las mantenían las organizaciones responsables de los ASs y enviaban info acerca de sus redes a las pasarelas empleando EGP.

3.2.1.2 Pasarelas nucleares

Además de la detección y reporte de mensajes de ICMP, el núcleo de ARPANET también implementaba:

- GGP("Gateway-to-Gateway Protocol") para intercambiar información de conectividad entre pasarelas nucleares(ver

Gateway-to-Gateway Protocol (GGP)).

- EGP("Exterior Gateway Protocol ") para recoger información de conectividad de pasarelas no nucleares(ver Exterior Gateway Protocol (EGP)).
- XNET("Cross-Network Debugging Protocol"), usado para cargar la pasarelas y para examinar sus datos.
- HMP("Host Monitoring Protocol") usado para recoger medidas y estadísticas de las pasarelas(RFC 869).

3.2.1.3 Pasarelas no nucleares

Las redes locales creaas por grupos individuales pueden generar múltiples redes físicas, unidas por pasarelas(no nucleares). Tal grupo de redes se denomina AS. Entre sus responsabilidades, un AS debe:

- Recoger información de accesibilidad de todas las redes conectadas.
- Anunciar la accesibilidad de la información al núcleo por medio de un protocolo externo.
- Tener un punto de contacto administrativo y técnico único.

Un AS debe recoger información de encaminamiento y accesibilidad acerca de sus propias redes internas. Las máquinas escogidas deben enviar esa información a otros Ass y a las pasarelas nucleares. Como se indicó arriba, se debe usar EGP para este comunicación inter-AS. Se puede usar cualquier IGP que se adapte a ello, siendo los dos más comunes:

- El prot descrito en El prot Hello
- RIP("Routing Information Protocol") descrito en RIP("Routing Information Protocol") Version 1

3.2.1.4 GGP("Gateway-to-Gateway Protocol")

GGP es un prot *histórico*. Su status es *no recomendado*. Se describe con detalle en el RFC *RFC 823 - La pasarela DARPA*.

Como se mencionó antes(ver Pas nucleares y no nucleares), las pasarelas nucleares originales de ARPANET usaban GGP para intercambiar info de encaminamiento. Además, tenían que encaminar datagramas que pasaran a través del núcleo . Cualquier datagrama en tránsito por el núcleo debería pasar por dos pasarelas nucleares. Los principios básicos de GGP son:

Cuando una pasarela nuclear se activa, se le asignan v*ecinos del* núcleo. Una pasarela sólo necesita propagar información sobre las redes de las que puede dar acceso a sus vecinos. Los vecinos actualizarán su info de encaminamiento y enviarán los cambios a sus vecinos asignados.

La info consiste en tuplas (N,C) donde:

N
: Es una red alcanzable sólo por esta pasarela

C
: Es el coste de alcanzar esa red. Se expresa en saltos de red(número de pasarelas a pasar). Un coste cero corresponde a una red conectada directamente al núcleo. El coste máximo corresponde a redes inalcanzables.

Los mensajes GGP se trasportan sobre datagramas IP, y típicamente contienen una lista de pares (N,C). Se envía por una pasarela a sus vecinos cuando ocurre uno de los sucesos siguientes:

- Una nueva red se alcanzable desde la pasarela.
- Un red se vuelve inalcanzable.
- Los datos de encaminamiento cambian debido a la recepción de mens GGP de sus pasarela vecinas.
- A la recepción de un mensaje GGP de la pasarela G, la pasarela vecina A comparará el par entrante (N, C) con el par (N, C) de sus tablas locales. Si el coste para alcanzar la red N es menor al usar la pasarela G(originadora del men GGP) que usando la información de encaminamiento almacenada en la tabla local, la ruta para la red N se actualiza para que apunte a la pasarela G, y como es un cambio, A generará un men GGP para informar a sus vecinos del cambio. Eventualmente, la info de la red N alcanzará todas las pasarelas del núcleo.

3.2.2 Arquitectura de encaminamiento NSFNET

Como se describe en NSFNET, la troncal NSFNET se ha implementado en tres fases y sus prots y arquitectura de encaminamiento

han evolucionado de forma consecuente. Esta evolución y las posibles alternativas para el futuro se describen con detalle en:

- *RFC 1074 -El SPFBIGP("Based Interior Gateway Protocol") de la troncal NSFNET*.
- *RFC 1092 - EGP y PBR("Policy Based Routing") en la nueva troncal NSFNET*.
- *RFC 1093 - La ar de enc NSFNET*.
- *RFC 1104 - Modelos de PBR("Policy Based Routing")*.
- *RFC 1133 - Encaminamiento entre NSFNET y DDN*.
- *RFC 1222 - Desarrollando la ar de enc NSFNET*.

La **primera troncal** usaba el protocolo Hello(ver El protocolo Hello) para el encaminamiento interior. Las redes de los clientes usaban en su mayoría RIP (ver RIP("Routing Information Protocol, versión 1")) como un IGP y se conectaban a la troncal empleando una interfaz entre Hello y RIP.

La **segunda troncal** utilizaba un subconjunto del prot de enc IS-IS del OSI ANSI ("Intermediate System to Intermediate System", ver IS-IS("OSI Intermediate System to Intermediate System ")) como IGP. Empleaba EGP (ver EGP("Exterior Gateway Protocol")) para intercambiar información de accesibilidad entre la troncal y las redes conectadas. El encaminamiento se controla con una base de datos de política de *encaminamiento distribuida* que controla la aceptación y distribución de info de encaminamiento. Esta base de datos la gestiona el NOC("Network Operations Center") y está disponible a través de los IS("Information Services").

En la **tercer troncal**, EGP fue reemplazado progresivamente por un verdadero protocolo de encaminamiento inter-AS llamado BGP("*Border Gateway Protocol"),* descrito en BGP("Border Gateway Protocol"). Un aspecto importante de BG^P es el tratamiento de Internet como un conjunto dde sistemas autónomos conectados arbitrariamente y sin núcleo. Esto elimina el requerimiento de que una sola red como NSFNET juegue un papel central y permita que la troncal esté compuesta de muchas redes.

 Tabla de contenidos  Protocolo de encaminamiento interior

3.3 Protocolos de encaminamiento interior

Los IGPs("*Interior routing protocols"* o "i*nterior gateway protocols")* se utilizan para intercambiar información de encaminamiento entre "routers" con un sólo sistema AS("autonomous system"). También lo usan los "routers" que ejecutan protocolos de encaminamiento exterior para recoger información de accesibilidad de la red para el AS.

Los IGPs más usados son:

- El protocolo Hello(ver El protocolo Hello).
- RIP("Routing Information Protocol")(ver RIP("Routing Information Protocol")).
- El protocolo OSPF("Open Shortest Path First")(ver OSPF("Open Shortest Path First Protocol") Versión 2).

Antes de discutir en detalle estos tres protocolos, observaremos dos grupos importantes de algoritmo de encaminamiento usados en IGPs.

3.3.1 Algoritmos de encaminamiento

En esta sección se discuten los algoritmos de encaminamiento Vector-Distancia, Estado del Enlace, y el del Camino Más Corto.

3.3.1.1 Vector-Distancia

El término *Vector-Distancia se refiere* a una clase de algoritmos que usan las pasarelas para actualizar su información de encaminamiento. Cada "router" comienza con un conjunto de rutas para aquellas con las que está directamente conectado, y posiblemente algunos "routers" adicionales a otras redes o hosts si la topología de la red es tal que el protocolo de encaminamiento no es capaz de producir el encaminamiento deseado. Esta lista se guarda en una *tabla de encaminamiento,* en la que cada entrada identifica una red o host de destino y a la "distancia" a ella. Este distancia se denomina *métrica* y se mide típicamente en saltos.

Periódicamente, cada "router" envía una copia de su tabla de encaminamiento a cualquier otro "router" que pueda alcanzar directamente. Cuando un informe le llega al "router" B del A, B examina el conjunto de destinos que recibe y la distancia a cada uno. B actualizará su tabla de encaminamiento si:

- A conoce un camino más corto a cada destino.
- A lista un destino que B no tiene en su tabla.
- La distancia de A a un destino desde B pasando por A ha cambiado.

Esta clase de algoritmo es fácil de implementar, pero tiene un número de desventajas:

- Cuando las rutas cambian rápidamente, es decir, aparece una nueva conexión o falla una vieja, la topología de encaminamiento puede no estabilizar la topología cambiada debido a que la información se propague lentamente y mientras se esté propagando, algunos "routers" tengan información de encaminamiento incorrecta.

  Otra desventaja es que cada "router" tiene que enviar una copia de toda su tabla de encaminamiento a cada vecino a intervalos regulares. Por supuesto, se pueden usar intervalo más largos para reducir la carga de la red pero eso introduce problemas relacionados con la respuesta de la red a cambios en la topología.

  Los algoritmos vector-distancia que usan la cuenta de saltos como métrica no tienen en cuenta la velocidad o la fiabilidad del enlace.

La tarea más difícil en uno de estos algoritmos es prevenir la inestabilidad. Existen distintas soluciones:

- **Cuenta hasta infinito**

  Elijamos un valor de 16 para representar infinito. Suponer que una red se vuelve inestable; todos sus vecinos generan un timeout y fijan la métrica de esa red a 16. Podemos considerar que todos los "routers" vecinos tienen algún elemento hardware que los conecta con la red desaparecida, con un coste de 16. Ya que se trata de su única conexión con esa red, el resto de los "router" convergerá hacia nuevas rutas que pasen por los vecinos con una conexión directa pero no disponible. Una vez que se ha producido la convergencia, todos los "routers" tendrán una métrica de16 para la red desaparecida. Como 16 indica infinito, la red será inaccesible para todos.

  La cuestión con los algoritmos vector-distancia no es si se producirá la convergencia, sino cuánto tiempo llevará. Consieremos la configuración mostrada en Figura - El problema de la cuenta hasta infinito.

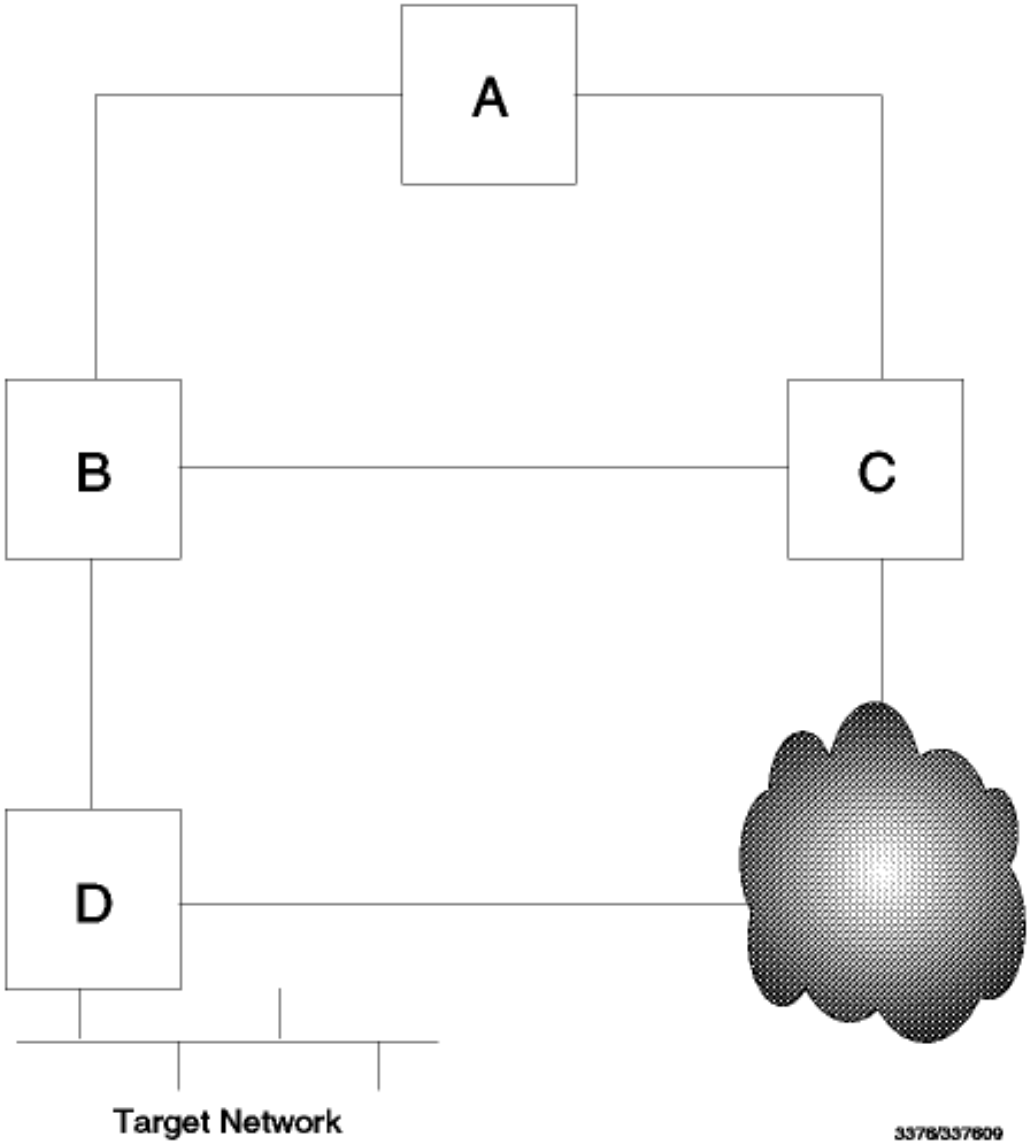


Figura: El problema de la cuenta hasta infinito - Todos los enlaces tienen una métrica de 1 excepto por la ruta indirecta de C a D que tiene métrica de 10.

Considerar ahora sólo las rutas de cada pasarela a la red de destino.

Gateway	First Hop	Metric
D	=	1
B	D	2
C	B	3
A	B	3

Considerar ahora que el enlace de B a D falla. Las rutas se deberían ajustar para utilizar el enlace de C a D. Los cambios en el encaminamiento comienzan cuando B se da cuenta de que la ruta D ya no es utilizable. En RIP esto sucede cuando B no recibe actualización de su enlace durante 180 segundos.

La siguiente imagen muestra la métrica para la red de destino, como aparece en la tabla de encaminamiento de cada pasarela.

Time	⟶						....				
GW	First Hop	Metric	FH	M	FH	M	.....	FH	M	FH	M
D	=	1	=	1	=	1	.....	=	1	=	1
B	Unreach.	Unreach.	C	4	C	5	.....	C	11	C	12
C	B	3	A	4	A	5	.....	A	11	D	11
A	B	3	C	4	C	5	.....	C	11	C	12
	Iteration #1		#2		#3			#9		#10	

= : directly connected
Unreach: unreachable
FH : First hop
M : Metric

Figura: El problema de la cuenta hasta infinito

El problema es que B se puede liberar de su ruta a (con un mecanismo de timeout), pero quedan vestigios de esa ruta en el sistema durante mucho tiempo(el tiempo entre iteraciones es de 30 segundos en RIP). Inicialmente, A y C todavía piensan que pueden alcanzar a D vía B, así que siguen enviando actualizaciones con métricas de 3. B las recibirá y, en la siguiente iteración, dirá que puede llegar a D por A o por C. Por supuesto, no puede, ya que las rutas dadas por A y C(D alcanzable vía B con métrica 3) ya no están, pero no hay forma de saberlo aún. Incluso cuando descubren que sus rutas por B han desaparecido, todavía piensan que hay una ruta disponible a través del otro. Al final el sistema convergerá, cuando el enlace directo de C a D tenga menor coste que el recibido(por C) de B a A. El peor caso es cuando una red se vuelve completamente inaccesible desde alguna parte del sistema: en este caso, las métricas se pueden incrementar lentamente de modo parecido al indicado arriba hasta que finalmente alcancen el valor infinito. Por esta razón, el problema se llama *cuenta hasta infinito.* De esta forma, la elección de la "infinidad" es un compromiso entre el tamaño de la red y la velocidad de convergencia en el caso de que la cuenta se produzca. Esto explica por qué se elige una valor tan bajo como16, que es el usado por RIP.

- Las otras soluciones se discutirán en el protocolo RIP(ver RIP("Routing Information Protocol")).

3.3.1.2 Estado del enlace, Primero el Camino Más Corto

El crecimiento de las redes durante los últimos años ha forzado los *IGPs* más allá de sus límites. La alternativa primaria a los esquemas vector-distancia es una clase de protocolos conocida como *Estado del enlace, Primero el Camino Más Corto*.

Sus características principales son:

- Un conjunto de redes físicas se divide en un número de áreas.
- Todos los "routers" dentro de un área tiene idénticas bases de datos.
- La base de datos de cada "router" describe la topología completa del dominio de encaminamiento(qué "routers" se conectan a qué redes). La topología de un área se representa en una base de datos denominada *LSD("Link State Database")* que es una descripción de todos los enlaces de los "routers" del área.
- Cada "router" usa su base de datos para derivar el conjunto de caminos mínimos a todos los destinos de los que construye su tabla. El algoritmo usado para determinar los caminos óptimos se llama *SPF("Shortest Path First")*.

En general, un protocolo enlace estado trabaja del modo siguiente. Cada "router" envía periódicamente una descripción de su conexión(el estado de su enlace) a sus vecinos(aquellos conectados a la misma red). Esta descripción, llamada LSA *("Link State Advertisement"),* incluye el coste de la conexión. El LSA inunda el dominio del "router". Cada "router" del dominio mantiene una copia idéntica y sincronizada de una base de datos compuesta de la información del estado del enlace. Esta base de datos describe la topología del dominio como las rutas a redes fuera del dominio como son rutas a redes en otros AS. Cada "router" ejecuta un algoritmo sobre su base de datos resultando en un árbol del camino mínimo(MST o "Minimun Spanning Tree"), que contiene la ruta más corta para cada "router" y red que pueda alcanzar la pasarela. A partir de él, el coste hasta el destino y el salto siguiente para retransmitir un dato se utilizan para construir la tabla de encaminamiento del "router".

Este tipo de protocolos, en comparación con los protocolos vector-distancia, envían actualizaciones cuando hay noticias, que pueden ser regulares para asegurar a los vecinos que la conexión sigue activa. Lo que es más importante, la información intercambiada es el estado del enlace, no los contenidos de la tabla de encaminamiento. Esto significa que los algoritmos del estado del enlace usan menos recursos que su contrapartida vector-

distancia, sobre todo cuando el encaminamiento es complejo o el AS grande. Sin embargo, tienen un elevado coste computacional. A cambio, los usuarios consiguen respuesta a los eventos de red más rápido, convergencia más veloz, y acceso a servicios de red más avanzados.

3.3.2 El protocolo Hello

Se usó en el software "Fuzzball" para minicomputadores LSI/11, ampliamente usados en la experimentación en Internet. Se describe en el *RFC 891 - Protocolos LAN DCN*. No es un est de Internet.

Nota: OSPF (ver OSPF("Open Shortest Path First Protocol") Versión 2) incluye un protocolo muy diferente para negociación entre "routers" llamado también Hello.

La comunicación en el protocolo Hello se hace por mensajes Hello sobre datagramas IP. El número de protocolo de Hello es el 63(reservado para "cualquier red local").

El protocolo Hello es significativo parcialmente debido a su amplia distribución durante la expansión de Internet y parcialmente porque es un ejemplo de algoritmo vector-distancia que no usa cuenta de saltos como RIP(ver RIP("Routing Information Protocol") Versión 1) sino retardos de red como métrica.

Un host físico *DCN*("*Distributed Computer Network")* es un procesador compatible con el PDP11 que soporta un número de procesos cooperativos secuenciales, a cada uno de los cuales se le da un id unívoco de 8 bits llamado su ID de puerto. Cada host DCN contiene uno o más procesos de Internet , cada uno de los cuales soporta un host virtual dado un ID de 8 bits, llamado el ID de host. Existe una correspondencia uno a uno entre las direcciones de Internet y los IDs de hosts. Todos los host físicos DCN se identifican por su ID de host con el fin de detectar bucles al actualizar tablas, que establecen los caminos de mínimo retardo entre los host virtuales.

Cada host físico tiene dos tablas:

Tabla de Host
: Contiene estimaciones del retardo del viaje de ida y vuelta y un desplazamiento lógico de reloj(es decir, la diferencia entre el reloj lógico de este host y el del emisor). Se indexa por el número de host. Se mantiene dinámicamente mediante actualizaciones generadas por mensajes Hello periódicos(de 1 a 30 segundos).

Tabla de red
: Contiene una entrada para cada red vecina conectable a la red local y a otras redes concretas que no sean vecinas. Cada entrada contiene el número de red, además del número de host del "router"(localizado en la red local) para esa red. Esta tabla se inicializa en tiempo de configuración para todos los host excepto en aquellos que soporten los protocolos de encaminamiento GGP o EGP. En estos casos, se actualiza como parte de la operación de encaminamiento.

Además, las entradas de ambas tablas las pueden cambiar los comandos del operador.

El retardo y el desplazamiento estimados son actualizados por mensajes Hello intercambiados en los enlaces que conectan los vecinos físicos.

Este es el formato de un mensaje Hello:

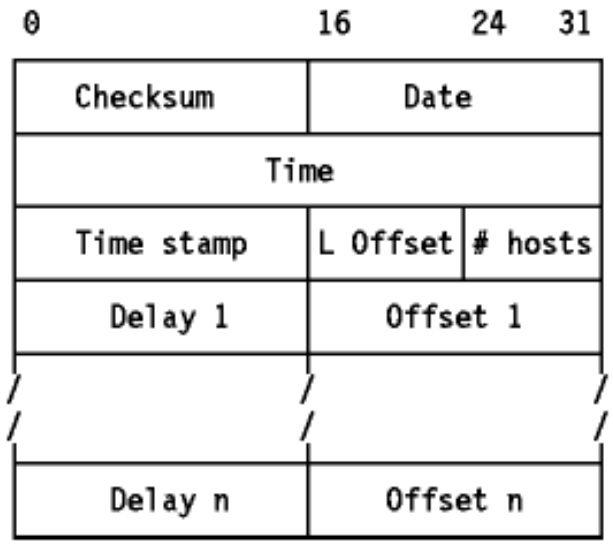


Figura: Formato del mensaje Hello

Donde:

Checksum
: contiene un checksum cubriendo los campos indicados

Date
: es la fecha local del host

Time

es la hora local del host

Timestamp
: Usado en cálculos del tiempo de viaje

L Offset
: contiene el desplazamiento del bloque de entradas de direcciones de Internet usado en la red local

#hosts
: contiene el número de entradas de la tabla de host siguiente

Delay n
: Retardo hasta el host n

Offset n
: offset para el host n (diferencia entre los relojes)

Consideremos ahora los dos pasos principales del protocolo Hello.

3.3.2.1 Cálculo del retardo del viaje

Periódicamente cada host envía un mensaje Hello a su vecino en cada enlace común. Para cada uno de estos enlaces el emisor guarda un conjunto de variables de estado, incluyendo una copia de del campo dirección fuente del último mensaje Hello recibido. Al construir un mensaje Hello el emisor fija el campo de destino a su variable de estado y el de dirección fuente su propia dirección. Luego rellena los campo s fecha y hora a partir de su reloj y el sello de tiempo de otra variable de estado. Finalmente copia el retardo y los valores de offset de su tabla de host al mensaje.

Los cálculos del retardo del viaje se realizan cuando el host recibe el mensaje. Cada enlace tiene asignada una variable interna de estado, que se actualiza a la recepción de cada mensaje Hello; esta variable toma el valor del campo hora, menos la hora actual de ese momento. Cuando se transmite el siguiente mensaje Hello, el valor asignado al campo sello de tiempo se computa como los 16 bits de oren inferior de esta variable menos la hora actual. El retardo se calcula como los 16 bits de orden inferior de la hora actual menos el valor del sello de tiempo.

3.3.2.2 Actualizaciones del host

Cuando llega un mensaje Hello, lo que da lugar al cálculo de un retardo de viaje, se efectúa un proceso de actualización del host. Consiste en añadir el retardo a cada una de las n entradas de retardos en el mensaje Hello y en comparar cada uno de esos valores con el campo retardo("delay") de la tabla de host correspondiente. Cada entrada se actualiza según las siguiente reglas:

- Si el enlace conecta con otro host en la misma red y el ID de puerto del proceso de salida del enlace coincide con el campo ID de puerto de la entrada, se actualiza la entrada.
- Si el enlace conecta a otro host en la misma red y el ID de puerto del proceso de salida del enlace no coincide con el número de puerto de la entrada *y el* retardo calculado es menor que el campo de retardo del host de la tabla de host en al menos un umbral de conmutación especificado(habitualmente 100ms), se actualiza la entrada. Por ejemplo, si el host A envía a B un mensaje Hello, y el retardo actual de B para alcanzar es mayor que el retardo de A a D más el retardo de B a A, B cambia su ruta y envía el tráfico a D por A.

El propósito del umbral de conmutación es evitar(además de ser una especificación del retardo mínimo) conmutaciones innecesarias entre enlaces y bucles transitorios que pueden ocurrir debido a variaciones normales en los retardos de propagación.

Remitirse al RFC 891 para más detalles.

3.3.3 RIP("Routing Information Protocol")

Hay dos versiones de RIP. La versión 1, llamada RIP a secas, es un protocolo muy extendido con un número de limitaciones conocidas. La versión 2 es una versión mejorada diseñada para aliviar estas limitaciones siendo al mismo tiempo muy compatible con su predecesor.

3.3.3.1 RIP("Routing Information Protocol" Versión 1)

RIP es un *protocolo est* (STD 34). Su status es *electivo*. Se describe en el RFC 1058, aunque muchas implementaciones de RIP datan de años atrás a este RFC. RIP se implementa con un "demonio" llamado *"routed".* También soportan RIP los "demonios" de tipo *gated*.

RIP se basa en los protocolos de encaminamiento PUP y XNS de Xerox PUP. Es muy usado, ya que el código está incorporado en el código de encaminamiento del BSD UNIX que constituye la base para muchas implementaciones de UNIX.

RIP es una implementación directa del encaminamiento vector-distancia para LANs. Utiliza UDP como protocolo de transporte, con el número de puerto 520 como puerto de destino(ver UDP("User Datagram Protocol") para una descripción de UDP y de los puertos). RIP opera en uno de dos modos: *activo* (normalmente usado por "routers") y *pasivo* (normalmente usado por hosts). La diferencia entre los dos se explica más abajo. Los mensajes RIP se envían en datagramas UDP y cada uno contiene hasta 25 pares de números como se muestra en Figura - MensajesajeRIP.

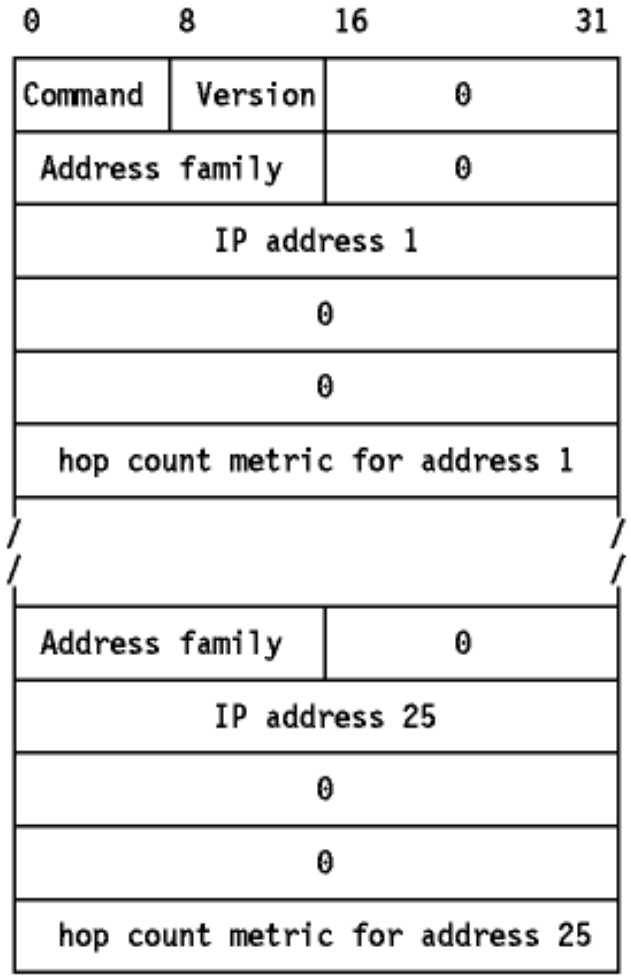


Figura: Mensaje RIP - En un mensaje RIP se pueden listar entre 1 y 25 rutas. Con 25 rutas el mensaje tiene 504 bytes(25x20+4) que es el tamaño máximo que se puede transmitir en un datagrama UDP de 512 bytes.

Command
: es 1 para una petición RIP o 2 para una respuesta.

Version
: es 1.

Address Family
: es 2 para direcciones IP.

IP address
: es la dirección IP de para esta entrada de encaminamiento: un host o una subred(caso en el que el número de host es cero).

Hop count metric
: es el número de saltos hasta el destino. La cuenta de saltos para una interfaz conectada directamente es de 1, y cada "router" intermedio la incrementa en 1 hasta un máximo de 15, con 16 indicando que no existe ruta hasta el destino.

Tanto el modo activo como el pasivo escuchan todos los mensajes de broadcastadcast y actualizan su tabla de encaminamiento según el algoritmo vector-distancia descrito antes.

Operaciones básicas

- Cuando RIP se inicia envía un mensaje a cada uno de sus vecinos(en el puerto bien conocido 520) pidiendo una copia de la tabla de encaminamiento del vecino. Este mensaje es una solicitud(el campo "command" se pone a 1) con "address family" a 0 y "metric" a 16. Los "routers" vecinos devuelven una copia de sus tablas de encaminamiento.
- Cuando RIP está en modo activo envía toda o parte de su tabla de encaminamiento a todos los vecinos(por broadcastadcast y/o con enlaces punto a punto. Esto se hace cada 30 segundos. La tabla de encaminamiento se envía como respuesta("command" vale 2, aun que no haya habido petición).
- Cuando RIP descubre que una métrica ha cambiado, la difunde por broadcastadcast a los demás "routers".
- Cuando RIP recibe una respuesta, el mensaje se valida y la tabla local se actualiza si es necesario.

  Para mejorar el rendimiento y la fiabilidad, RIP especifica que una vez que un "router"(o host) a aprendido una ruta de otro, debe guardarla hasta que conozca una mejor(de coste estrictamente menor). Esto evita que los "routers" oscilen entre dos o más rutas de igual coste.

- Cuando RIP recibe una petición, distinta de la solicitud de su tabla, se devuelve como respuesta la métrica para cada entrada de dicha petición fijada al valor de la tabla local de encaminamiento. Si no existe ruta en la tabla local, se pone a 16.
- Las rutas que RIP aprende de otros "routers" expiran a menos que se vuelvan a difundir en 180 segundos(6 ciclos de broadcastadcast). Cuando una ruta expira, su métrica se pone a infinito, la invalidación de la ruta se difunde a los vecinos, y 60 segundos más tarde, se borra de la tabla.

Limitaciones

RIP no está diseñado para resolver cualquier posible problema de encaminamiento. El RFC 1720 (STD 1) describe estas limitaciones técnicas de RIP como "graves" y el IETF está evaluando candidatos para reemplazarlo. Entre los posibles candidatos están OSPF("Open Shortest Path First Protocol" Versión 2) y el IS-IS de OSI IS-IS (ver IS-IS("Intermediate System to Intermediate System" de OSI)). Sin embargo, RIP está muy extendido y es probable que permanezca sin sustituir durante algún tiempo. Tiene las siguientes limitaciones:

- El coste máximo permitido en RIP es 16, que significa que la red es inalcanzable. De esta forma, RIP es inadecuado para redes grandes(es decir, aquellas en las que la cuenta de saltos puede aproximarse perfectamente a 16).
- RIP no soporta máscaras de subred de longitud variable*(variable subnetting)*. En un mensaje RIP no hay ningún modo de especificar una máscara de subred asociada a una dirección IP.
- RIP carece de servicios para garantizar que las actualizaciones proceden de "routers" autorizados. Es un protocolo inseguro.
- RIP sólo usa métricas fijas para comparar rutas alternativas. No es apropiado para situaciones en las que las rutas necesitan elegirse basándose en parámetros de tiempo real tales como el retardo, la fiabilidad o la carga.
- El protocolo depende de la *cueanta hasta infinito* para resolver algunas situaciones inusuales. Como se describió antes(Vector-Distance), la resolución de un bucle requeriría mucho tiempo(si la frecuencia de actualizaciones fuese limitada) o mucho ancho de banda(si las actualizaciones se enviasen por cada cambio producido). A medida que crece el tamaño del dominio, la inestabilidad del algoritmo vector-distancia de cara al cambio de topología se hace patente. RIP especifica mecanismos para minimizar los problemas con la cuenta hasta infinito(descritos más abajo) que permiten usarlo con dominios mayores, pero eventualmente su operatividad será nula. No existe un límite superior prefijado, pero a nivel práctico este depende de la frecuencia de cambios en la topología, los detalles de la topología de la red, y lo que se considere como un intervalo máximo de tiempo para que la topología de encaminamiento se estabilice.

La resolución de la *cuenta hasta infinito* se efectúa usando las técnicas *split horizon*, *poisoned reverse* y *triggered updates*.

Split horizon con poisoned reverse

Consideremos nuestra red de ejemplo(mostrada en Figura - El problema de la cuenta hasta infinito) de nuevo.

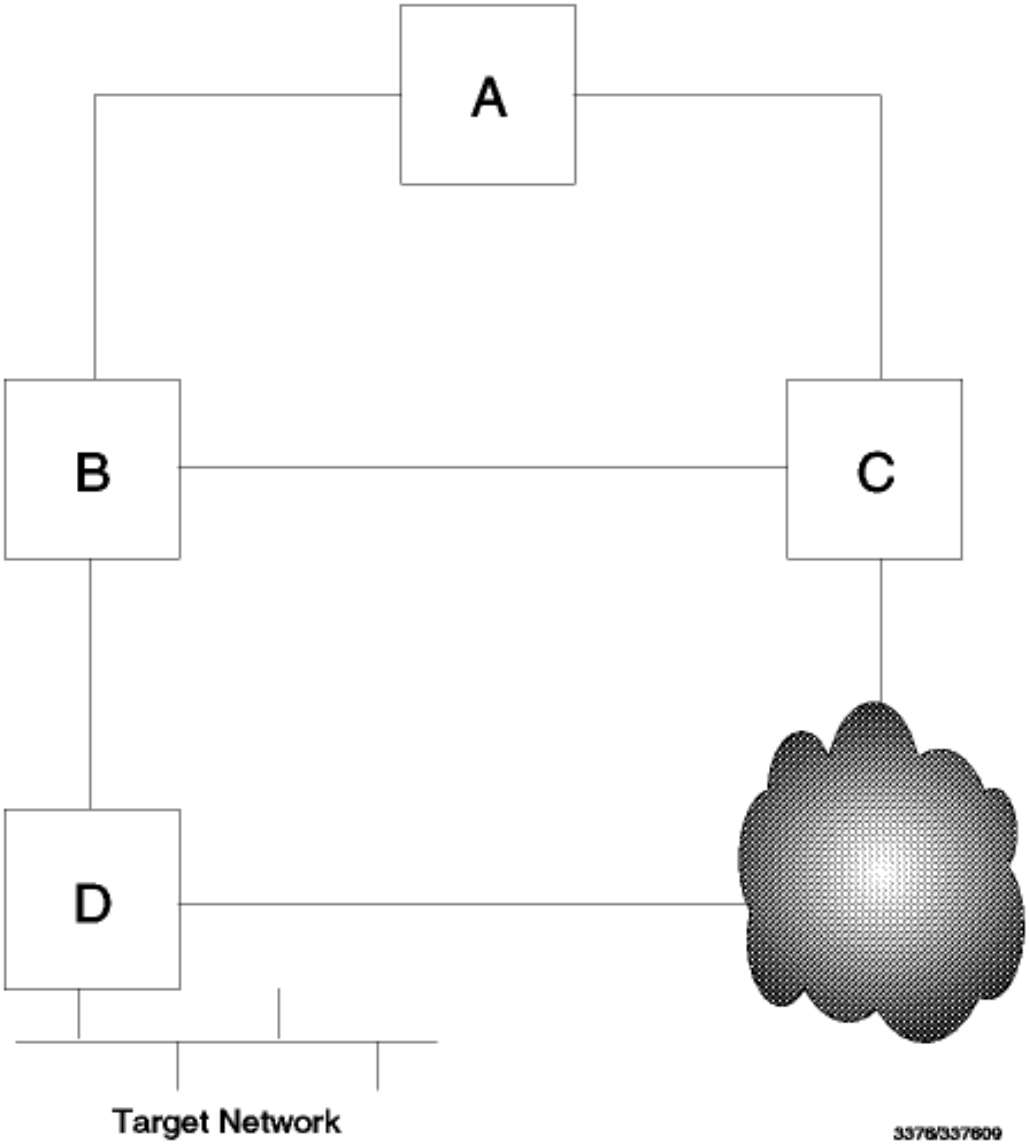


Figura: El problema de la cuenta hasta inifinito - Todos los enlaces tienen una métrica de 1 excepto por la ruta indirecta de C a DD que tiene una métrica de 10.

Como se describió en Vector-Distancia, el problema lo causaba el hecho de que A y C se decepcionan mutuamente. Cada uno afirma ser capaz de alcanzar a D a través del otro. Este hecho se puede evitar siendo más cuidadoso con el destino de la información. En particular, nunca es útil afirmar la accesibilidad a una red de destino a través del vecino del que se aprendió la ruta. El método *split horizon con poisoned reverse* incluye las rutas en las actualizaciones enviadas al "router" el que se aprendieron, pero pone sus métricas a infinito. Si dos "routers" tienen rutas apuntándose recíprocamente, el anunciar las rutas en bucle con métrica de 16 romperá el bucle inmediatamente. Si simplemente no se anuncian estas rutas(esquema conocido como *simple split horizon*), las rutas erróneas tendrán que ser eliminadas tras un timeout. Poisoned reverse tiene una desventaja: incrementa el tamaño de los mensajes de encaminamiento.

Triggered updates

Split horizon con poisoned reverse evita cualquier bucle que implique sólo dos pasarelas. Sin embargo, aún es posible acabar con situaciones en las de este tipo. Por ejemplo, A puede creer que tiene una ruta a través de B, B a través de C, y C a través de A. Esto no se puede solucionar con el método split horizon. Este bucle sólo se arreglará cuando la métrica alcance infinito y la red o el host implicados se declaren inaccesibles. El método triggered updates es un intento de acelerar esta convergencia. Siempre que un "router" cambia la métrica de una ruta, se le requiere que envía mensajes casi inmediatamente, incluso aunque no sea el momento de una actualización(RIP especifica un pequeño intervalo, entre 1 y 5 segundos, con el fin de evitar que estas actualizaciones generen un tráfico de red excesivo).

3.3.3.2 RIP-2("Routing Information Protocol" Versión 2)

RIP-2 es un *borrador.* Su status es *electivo*. Se describe en el RFC 1723.

RIP-2 extiende RIP-1. Es menos potente que otros IGPs recientes tales como OSPF(ver OSPF("Open Shortest Path First Protocol") Versión 2) de IS-IS (ver IS-IS("Intermediate System to Intermediate System") de OSI)), pero tiene las ventajas de una fácil implementación y menores factores de carga. La intención de RIP-2 es proporcionar una sustitución directa de RIP que se pueda usar en redes pequeñas y medianas, en presencia de subnetting variable(ver Subredes) o supernetting (ver CIDR("Classless Inter-Domain Routing")) y, sobretodo, que pueda interoperar con RIP-1.

RIP-2 aprovecha que la mitad de los bytes de un mensaje RIP están reservados(deben ser cero) y que la especificación original estaba diseñada con las mejoras en la mente de los desarrolladores, particularmente en el uso del campo de versión. Un área notable en la que este no es el caso es la interpretación del campo de métrica. RIP-1 lo especifica con un valor de 0 a 16 almacenado en un campo de 4 *bytes.* Por compatibilidad, RIP-2 preserva esta definición, lo que significa en que interpreta 16 como infinito, y desperdicia la mayor parte del rango de este campo.

Nota: Ni RIP-1 ni RIP-2 son adecuados para ser usados como IGPs en un AS en el que el valor de 16 sea demasiado bajo para ser considerado infinito, ya que lo valores altos del infinito exacerban el problema de la cuenta hasta infinito. El protocolo estado del enlace, más sofisticado, usado en OSPF y en IS-IS proporciona una solución de encaminamiento mucho mejor cuando el AS es lo bastante largo para tener una cuenta de saltos cercana a 16.

Si una implementación de RIP obedece la especificación RFC 1058, RIP-2 puede interoperar con ella. El formato del mensaje RIP-2 se muestra en Figura - MensajesajeRIP-2.

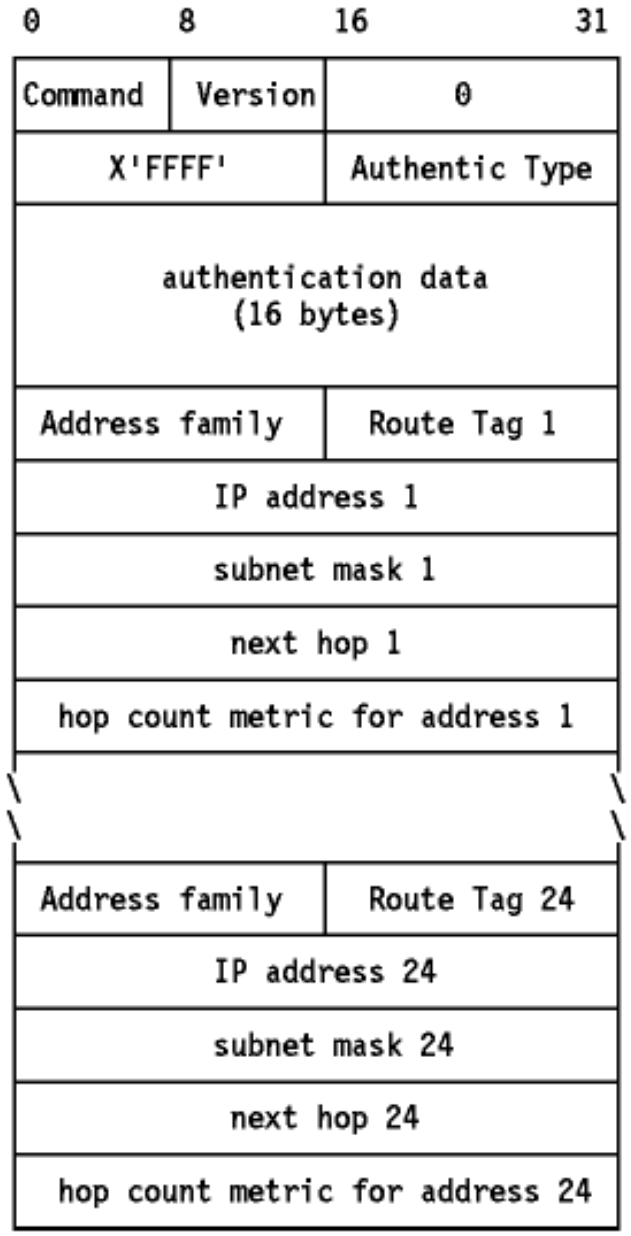


Figure: Mensaje RIP-2 - La primera entrada del mensaje puede ser una entrada de autentificación, como se muestra aquí, o una ruta como en el mensaje RIP. Si la primera entrada es de autentificación, sólo se pueden incluir 24 rutas en el mensaje; de otro modo, el máximo es 25, como en RIP.

Los campos del mensaje RIP-2 son los mismos que en RIP excepto los siguientes:

Version
: es 2. Le dice al "router" RIP-1 que ignore los campos reservados, los que deben ser cero(si el valor es 1, los "routers" deben desechar los mensaje con valores distintos de cero en estos campos, ya que los originó un "router" que dice ser RIP, pero que envía mensajes que no cumplen el protocolo).

Address Family
: puede ser X'FFFF' sólo en la primera entrada, indicando que se trata de una entrada de autentificación.

Authentication Type
: Define como se han de usar los restantes 16 bytes. Los únicos tipos definidos son 0, indicando ninguna autentificación, y 2 indicando que el campo contiene datos de password.

Authentication Data
: el password es de 16 bytes, texto ASCII plano, alineado a la izquierda y rellenado con caracteres nulos ASCII (X'00').

Route Tag
: es un campo dirigido a la comunicación de información acerca del origen de la información de encaminamiento. Está diseñado para la interoperabilidad entre RIP y otros protocolos de encaminamiento. Las implementaciones de RIP-2 deben conservarlo, aunque RIP-2 no especifica como se debe usar.

Subnet Mask
: la máscara de subred asociada con la subred a la que se refiere esta entrada.

Next Hop
: Una recomendación acerca del siguiente salto que el "router" debería usar para enviar datagramas a la subred o al host dado en la entrada.

Para asegurar una interoperabilidad segura con RIP, el RFC 1723 especifica las siguientes restricciones para los "routers" RIP-2 que transmiten sobre una interfaz de red en la que un "router" RIP puede escuchar y operar con mensajes RIP.

1. La información interna a una red nunca se debe anunciar a otra red.

2. La información acerca de una subred más específica no se debe anunciar donde los "routers" vean una ruta de host.
3. Las rutas a superredes(rutas con una máscara de subred más corta que la máscara natural de la red) no se deben anunciar en los sitios en los que puedan ser malentendidas por los "routers" RIP.

RIP-2 soporta además el multicast con preferencia al broadcastadcast. Esto puede reducir la carga de los host que no están a la escucha de mensajes RIP-2. Esta opción es configurable para cada interfaz para asegurar un uso óptimo de los servicios RIP-2 cuando un "router" conecta redes mixtas RIP-1/RIP-2 con redes RIP-2. Similarmente, el uso de la autentificación en entornos mixtos se puede configurar para adecuarse a los requerimientos locales.

RIP-2 está implementado en versiones recientes del "*gated* daemon", llamado con frecuencia "*gated Version 3"*. Ya que el borraddor es nuevo en el momento de su redacción, muchas implementaciones se ajustarán a la versión anterior descrita en el RFC 1388. Tales implementaciones interoperarán con aquellas que se adhieran al RFC 1723.

Para más información acerca de RIP-2, ver:

- *RFC 1721 - Análisis del protocolo RIP Versión 2*
- *RFC 1722 - Aplicabilidad del protocolo RIP Versión 2*
- *RFC 1723 - RIP Versión 2 - Transmisión de información adiciona*
- *lRFC 1724 - Extensión MIB de RIP Versión 2*

3.3.4 OSPF ("Open Shortest Path First Protocol") Versión 2

Nota: El término OSPF se usa invariablemente para referirse a OSPF Versión 2 (OSPF-2). OSPF Version 1, descrito en el RFC 1131, es obsoleto.

OSPF es un *borrador.* Su status es *electivo,* pero el RFC 1370 contiene condiciones de aplicabilidad para OSPF que dicen que cualquier "router" que implemente un protocolo distinto del encaminamiento IP simple debe implementar OSPF(lo que no significa que el "router" pueda implementar otros protocolos, por supuesto). OSPF se describe en el RFC 1583, que desfasa al RFC 1247. Las implementaciones basadas en el RFC 1583 son retrocompatibles con las basadas en el 1247 y pueden interoperar con ellas. Los apéndices F del RFC 1247 y E del 1583 hablan del desarrollo de OSPF 2 a partir de OSPF 1.

OSPF es un protocolo de encaminamiento interior, pero está diseñado para operar con un protocolo exterior adecuado, tal como BGP. Ver Interacción BGP OSPF.

OSPF es un est complejo en comparación con RIP: el RFC 1583 tiene 216 páginas, mientras que RIP, especificado en el RFC 1058 tiene 33 y RIP-2 (RFC 1723) 9 más. Mucha de su complejidad tiene un sólo propósito: asegurar que las bases de datos topológicas son las mismas para todos los "routers" dentro de un área. Debido a que la base de datos es la base para todas las decisiones de encaminamiento, si los "routers" tuvieran bases de datos independientes, podrían tomar decisiones mutuamente conflictivas.

OSPF se comunica por medio de IP(su número de protocolo es el 89). Es un protocolo de estado del enlace, primero el camino más corto, como se describe en Estado del enlace, primero el camino más corto. OSPF soporta distintas clases de redes tales como redes punto a punto, de broadcastadcast, como Ethernet y redes en anillo, y de no broadcastadcast, como X.25.

La especificación de OSPF hace uso de *máquinas de estado* para definir el comportamiento de los "routers" que siguen el protocolo. Cada aspecto del trabajo de un "router" importante para OSPF, como sus interfaces de red y sus vecinos, puede hallarse en uno de un número finito de estados(por ejemplo, un vecino puede estar en el estado "caído"). Hay una máquina para componente(por ejemplo, dos interfaces de red tiene diferentes máquinas) y el estado de uno es independiente del resto. Los estados posibles bastan para describir todas las posibles condiciones relevantes al protocolo, por lo que una máquina de estados está siempre en uno, y sólo uno, de sus posibles estados. Los cambios de estado se producen como resultado de *eventos*. Hay un conjunto finito de eventos para cada tipo de máquina que es suficiente para representar todas las posibles ocurrencias del protocolo. El comportamiento de una máquina en respuesta a un evento viene dado por todas las posibles combinaciones de estado y evento. Por ejemplo, si la máquina de estado para una interfaz de red experimenta un evento InterfaceDown (CaídaDeInterfaz), la máquina cambia incondicionalmente al estado down(caído). El evento InterfaceDown lo genera la implementación de OSPF cuando recibe una indicación de un protocolo de nivel inferior de que la interfaz no está operativa. Ver el RFC 1583 para un descripción completa de cada máquina, sus posibles estados y eventos y los cambios asociados a ellos.

Aquí hay algunas definiciones necesarias para entender la secuencia de operaciones descrita posteriormente en esta sección:

Area
: Conjunto de redes dentro de un sólo AS que se han agrupado juntas. La topología de un área permanece oculta al resto del AS, y cada área tiene una base de datos topológica separada. El encaminamiento en el AS se produce en dos niveles, dependiendo de si la fuente y el destino de un paquete están en la misma área(*intra-area routing*) o en áreas diferentes(*inter-area routing*).
  - ❏ El encaminamiento intra-area lo determina sólo la propia topología del área. Es decir, el paquete se encamina sólo a partir de información obtenida dentro del área; no se puede usar información de encaminamiento obtenida fuera de la misma.
  - ❏ El encaminamiento inter-area se hace siempre a través de la *troncal*.

  La división de un sistema autónomo en áreas permite una reducción significante en el volumen del tráfico de encaminamiento requerido para gestionar la base de datos en un AS grande.

Backbone
: El backbone o troncal consiste en aquellas redes no contenidas en ningún área, los "routers" conectados a estas, y los "routers" pertenecientes a múltiples áreas. La troncal debe ser contigua a nivel lógico. Si no es contigua físicamente, los componentes deben usar enlaces *virtuales*(ver más bajo). La troncal es responsable de la información de encaminamiento entre áreas. La troncal misma tiene las propiedades de un área; su topología está separada de las de otras áreas.

Area Border Router(ABR)

Un "router" conectado a múltiples áreas. Tiene una copia de la base de datos de cada área a la que está conectado. Siempre forma parte de la troncal, y son responsables de la propagación de la información de encaminamiento inter-area a las áreas a las que están conectados.

Internal Router(IR)

Un "router" que no es de tipo ABR.

AS Border Router(ASBR)

Un "router" que intercambia información de encaminamiento con "routers" pertenecientes a otros AS. Todos los "routers" de un AS conocen el camino al todos los "routers" de tipo "boundary". Un ASBR puede ser un ABR o un IR. No tiene que ser parte de la troncal.

Nota: La nomenclatura para este tipo de "router" varía. El RFC 1583, usa el término *AS Boundary Router*. Los RFCs 1267 y 1268, *Border Router* y *Border Gateway*. El RFC 1340, *AS Border Router*. En adelante, usaremos este último.

Virtual Link (VL)

Un VL o enlace virtual es parte de la troncal. Sus extremos son dos ABR que comparten un área no troncal. El VL se trata como un enlace punto a punto con métrica igual a la métrica intra-area entre los extremos. El encaminamiento a través del VL se hace usando encaminamiento intra-area normal.

Transit Area

Un área a través de la que se produce la conexión física de un VL.

Stub Area(SA)

Un área configurada para usar el encaminamiento por defecto para el encaminamiento inter-AS. Se puede configurar en los sitios donde hay un sólo punto de salida del área, o donde se puede usar cualquier salida sin preferencia por ninguna ruta. Por defecto, las rutas inter-AS se copian a todas las áreas, por lo que el uso de SAs puede reducir las necesidades de almacenamiento de los "routers" dentro de aquellas áreas donde hay definidas muchas rutas inter-AS.

Multiaccess Network

Una red física que soporta la conexión de múltiples "routers". Se asume que cada par de "routers" de tal red es capaz de comunicarse directamente.

Hello Protocol

La parte del protocolo OSPF usada para establecer un mantener relaciones vecinales. No es el protocolo Hello descrito en El protocolo Hello.

Neighboring routers

Dos "routers" que tienen interfaces a una red común. En redes multiacceso, los vecinos se descubren dinámicamente por medio del protocolo Hello.

Cada vecino se representa con una máquina de estado que describe la conversación entre este "router" y su vecino. A continuación se muestra un breve boceto del significado de los estados. Ver la sección siguiente para una definición de los términos *adyacencia* y *"router" designado*.

Down

Estado inicial de la conversación de un vecino. Indica que no ha habido información reciente recibida del vecino.

Attempt

Un vecino o una red no broadcastadcast parece estar en estado "down" y se debería intentar contactar con ella enviando paquetes Hello regulares.

Init

Se ha recibido recientemente u paquete Hello del vecino. Sin embargo, la comunicación bidireccional no se ha establecido aún con él(es decir, el propio "router" no aparece en el paquete Hello).

2-way

En este estado, la comunicación entre dos "routers" es bidireccional. Se pueden establecer adyacencias, y los vecinos en este estado o en uno superior se pueden elegir como "routers" designados(de backup o copia de seguridad).

ExStart

Los dos vecinos están a punto de crear una adyacencia.

Exchange

Los dos vecinos se dicen el uno al otro lo que tienen en sus bases de datos topológicas.

Loading

Los dos vecinos están sincronizaciónronizando sus bases de datos topológicas.

Full

Los dos vecinos son ahora totalmente adyacentes, y sus bases de datos están sincronizadas.

Varios eventos causan un cambio de estado. Por ejemplo, si un "router" recibe un paquete de un vecino en estado "down", el estado del vecino cambia a "init", y se inicia un contador de inactividad. Si el contador se dispara(es decir, no se reciben más paquetes OSPF antes de que expire) el vecino retorna al estado "down". Remitirse al RFC 1583 para una desscripción completa de los estados y de la información de los eventos que causan los cambios.

Adjacency(Adyacencia)

Una nueva relación formada entre vecinos seleccionados con el fin de intercambiar información de encaminamiento. *No* todos los pares de vecinos se vuelven adyacentes. En particular, no todos estos pares permanecen sincronizados. Si todos los vecinos tuvieran que estar sincronizados, el número de pares sincronizados en una red multiacceso tal como una LAN sería $n(n-1)/2$ donde n es el número de "routers" de la LAN. En redes grandes, el tráfico de sincronización inundaría la red, volviéndola inutilizable. El concepto de adyacencias se usa para limitar el número de pares sincronizados a 2n - 1, asegurando que el flujo de sincronización es manejable.

Link State Advertisement (LSA)

Se refiere al estado local del "router" o de la red. Esto incluye el estado de las interfaces y adyacencias del "router". El conjunto de LSAs de todos los "routers" y redes forma la base de datos topológica del área.

Flooding

El proceso de asegurar que cada LSA se pasa entre "routers" adyacentes para alcanzar a cada "router" del área. Es un procedimiento fiable.

Designated Router

Cada red multiacceso con al menos dos "routers" conectados, tiene un DR("Designated Router" o "router" designado). El DR genera un LSA para la red. Ya que todas las bases de datos de todos los "routers" están sincronizadas por medio de las adyacencias, el DR juega un papel central en el proceso de sincronización.

Backup Designated Router (BDR)

Con el fin de suavizar la transición a un nuevo DR, existe un BDR para toda red multiacceso. El BDR es además adyacente a todos los "routers" de la red, y se convierte en DR cuando el DR anterior falla. Debido a que ya hay adyacencias entre el BDR y el resto de los "routers", no hace falta crear nuevas cuando el BDR sustituye al DDR, acortando el tiempo de reemplazo considerablemente. El BDR se elige mediante

el protocolo Hello.

Interface
: La conexión entre un "router" una de sus redes. Cada interfaz tiene uno de estado asociada con él que se obtiene de los protocolos de nivel inferior y del mismo OSPF. Aquí se da una breve descripción de cada estado. Remitirse al RFC 1583 para más detalles, y para información sobre los eventos que causan el cambio de los estados.

Down
: La interfaz no está disponible. Es el estado inicial de la interfaz.

Loopback
: La interfaz realiza un loop back al "router". No se puede usar para el tráfico regular de datos.

Waiting
: El "router" está intentando determinar la identidad del DR o del BDR.

Point-to-Point
: La interfaz se conecta a una red punto a punto o es un VL. El "router" forma una adyacencia con el "router" en el otro extremo.
Nota: Las interfaces no necesitan direcciones IP. Como el resto de Internet prácticamente no tiene necesidad de ver las interfaces del "router" en el enlace punto a punto, sólo las interfaces con las otras redes, cualquier dirección IP para el enlace se requeriría sólo para la comunicación entre los dos "routers". Para ahorrar espacio de direcciones IP, los "routers" pueden pasarse sin direcciones IP en el enlace. Esto tiene el efecto de que los dos "routers" parezcan uno sólo, aunque esto no trae inconvenientes. Tal enlace se llama enlace *innumerado*.

DR Other
: La interfaz está en una red multiacceso pero el "router" no es el DR ni el BDR. El "router" forma adyacencias con el DR y el BDR.

Backup
: El "router" es el BDR. Se le ascenderá a DR si el DR actual falla. El "router" forma adyacencias con cualquier otro "router" de la red.

DR
: El "router" es el DR. Forma adyacencias con todos los demás "routers" de la red. Además debe originar un anuncia del enlace de red para el nodo.

Type of Service (TOS) metrics
: En cada tipo de anuncio de estado del enlace, se pueden anunciar distintas métricas para cada tipo de servicio IP("IP Type of Service"). Siempre se debe especificar una métrica para TOS 0(usada para el encaminamiento OSPF de paquetes de protocolo). Pueden definirse métricas para otros valores de TOS; si no se hace, se les supone iguales a los de TOS 0.

Link State Database
: También llamados *grafo dirigido o base de datos topológica.* Se crea a partir de los anuncios del estado del enlace que generan los "routers" del área.
Nota: el RFC 1583 usa el término LSD("Link State Database") con preferencia al de base de datos topológica. En esta sección se ha usado el segundo término, pero en lo que resta de ella, la llamaremos LSD.

Shortest-Path Tree
: Cada "router" ejecuta el algoritmo SPF("Shortest Path First") en el LSD para obtener su árbol mínimo. El árbol da la ruta a cualquier host o red de destino dentro del área. Se usa para construir la tabla de encaminamiento.
Nota: debido a que cada "router" ocupa un lugar diferente en la topología del área, la aplicación del algoritmo SPF da un árbol distinto para cada "router", aunque la base de datos sea idéntica.
Los ABR pueden ejecutar múltiples copias del algoritmo pero construyen una sola tabla de encaminamiento.

Routing table
: La tabla de encaminamiento contiene entradas para cada destino: red, subred o host. Para cada destino, hay información para uno o más tipos de servicios(TOS). Para cada combinación de destino y tipo de servicio, hay entradas de una o más rutas óptimas a emplear.

Area ID(AID)
: Un número de 32 bits que identifica un área particular. La troncal tiene un AID de cero.

Router ID(RID)
: Un número de 32 bits que identifica un "router" particular. Cada "router" dentro de una AS tiene un RID unívoco. Una posible implementación es usar como RID la menor dirección IP del "router".

Router Priority(RP)
: Un entero sin signo de 8 bits, configurable por medio de una interfaz que indica la selección del BDR. Un RP de cero indica que el "router" no se puede elegir como DR.

3.3.4.1 Descripción de las operaciones en OSPF

La secuencia básica de operaciones realizadas por los "routers" OSPF routers es:

1. Descubrir vecinos OSPF
2. Elegir el DR
3. Formar adyacencias
4. Sincronizar bases de datos
5. Calcular la tabla de encaminamiento
6. Anunciar los estados de los enlaces

Los "routers" efectuarán todos estos pasos durante su activación, y los repetirán en respuesta a eventos de red. Cada "router" debe ejecutar estos pasos para cada red a la que está conectado, excepto para calcular la tabla de encaminamiento. Cada "router" genera y mantiene una sola tabla de encaminamiento para todas las redes.

Las siguiente secciones describen estos pasos.

Descubriendo vecinos OSPF

Cuando los "routers" OSPF se activan, inician y mantienen relaciones con sus vecinos usando el protocolo Hello. El protocolo además asegura que la comunicación entre vecinos sea bidireccional. Los paquetes Hello se envían periódicamente al exterior por todas las interfaces de los "routers". La comunicación bidireccional se indica si el propio "router" aparece en el paquete Hello del vecino. En una red de broadcast, los paquetes Hello se envían por multicast; los vecinos se descubren luego dinámicamente. En redes no broadcast, cada "router" que sea un DR potencial tiene una lista de

todos los "routers" conectados a la red y enviará paquetes Hello a todos los demás DRs potenciales cuando su interfaz a la red sea operativa por primera vez.

La cabecera OSPF se describe en Figura - Cabecera del paquete OSPF.

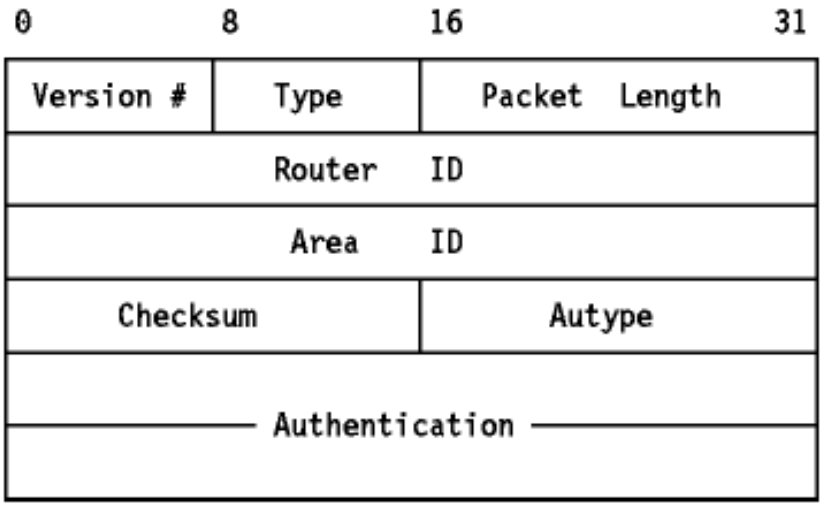


Figura: Cabecera del paquete OSPF

Version #
: El número de versión de OSPF (2).

Type
: El tipo: Hello (1), descripción de la base de datos (2), Link-State Request (3), Link-State Update (4), o Link-State Acknowledgment (5).

Packet length
: Longitud del paquete en bytes incluyendo la cabecera OSPF.

Router ID
: El ID del "router" que originó el paquete.

Area ID
: El área a la que se está enviando el paquete.

Checksum
: El checksum IP est de todo el contenido del paquete, excluyendo el campo de autentificación de 64 bits.

AuType
: Identifica el esquema de autentificación a usar con el paquete. El tipo de autentificación es configurable en por áreas. Los tipos de autentificación definidos actualmente son: 0 (ninguna autentificación) y 1 (password de 64 bits en texto plano).

Authentication
: Un campo de 64 bits usado para la autentificación.

El formato del paquete Hello de OSPF Hello se explica en Figura - El paquete Hello de OSPF.

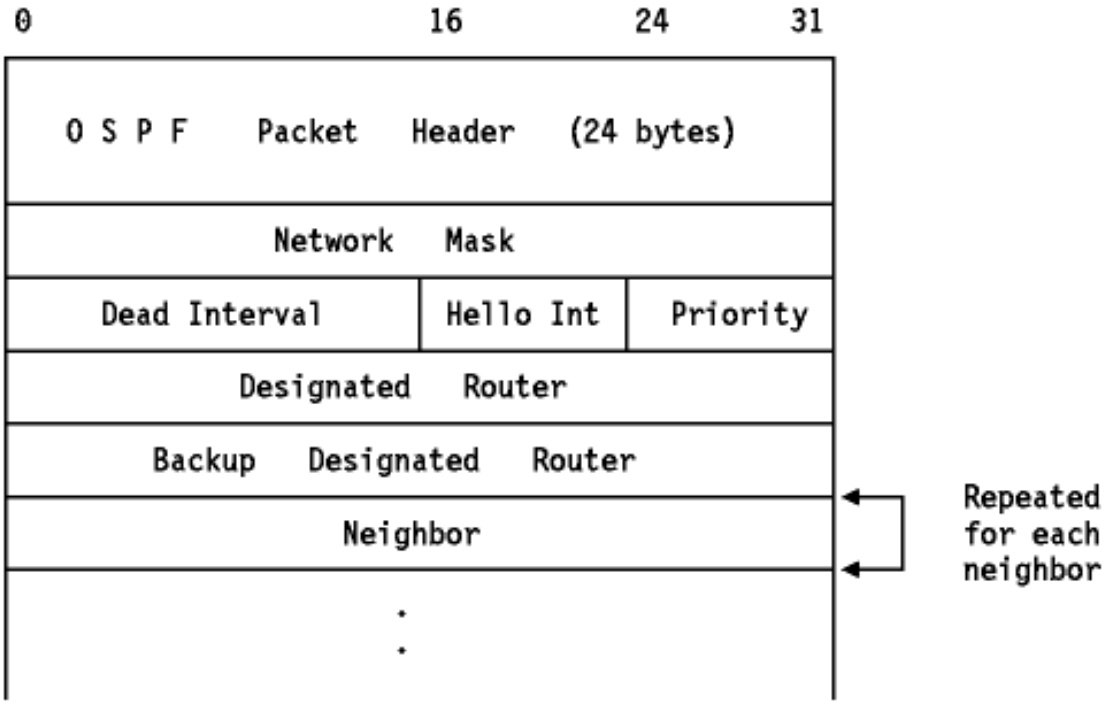


Figura: El paquete Hello de OSPF

Network Mask
: La máscara de red asociada a la interfaz. Se trata de la máscara de subred, si está implementada, o de su equivalente en una red sin subredes(por ejemplo, 255.255.255.0 para una red clase C si subredes).

Dead Interval
: El número de segundos que deben pasar antes de hacer que un vecino que permanece en silencio pase al estado "down".

Hello Int (Hello Interval)
: El número de segundos entre los paquetes Hello enviados por este "router".

Priority
: La RP("Router Priority") de este "router"(para esta interfaz).

Designated Router
: La dirección IP del DR de esta red, según el "router" emisor. Se pone a cero si no se conoce ningún DR.

Backup Designated Router
: La dirección IP del BDR de esta red, según el "router" emisor. Se pone a cero si no se conoce ningún BDR.

Neighbor
: El RID de cada "router" el que se han recibido paquetes Hello válidos recientemente. Recientemente significa dentro del último DI(Dead Interval).

Determinando el DR

Esto se hace usando el protocolo Hello. Aquí se da una breve descripción del proceso. Ver el RFC 1583 para una descripción completa. El "router" examina la lista de sus vecinos, desecha cualquiera que no tenga comunicación bidireccional o que tenga un RP de ver, y graba el DR, el BDR y la RP que ha declarado cada uno de ellos. El "router" se añade él mismo a la lista, usando el valor RP configurado para la interfaz y cero(desconocido) para el DR y el BDR, en el caso de que esté en proceso de activación.

Se emplean las siguiente reglas para determinar el BDR:

- Si uno o más "routers" declaran ser el BDR y no el DR, gana el que tenga un RP superior.
- En caso de empate, gana el que tenga mayor RID.
- Si ningún "router" declara ser el BDR, entonces el se elige el "router" con mayor RP a menos que se haya declarado como DR.
- De nuevo, en caso de empate gana el "router" con mayor RID.

Como el propio "router" que hace los cálculos está en la lista, puede determinar que él mismo es el BDR. Un proceso similar se sigue para el DR:

- Si uno o más "routers" declaran ser el DR, gana el que tenga un RP superior.
- En caso de empate, gana el que tenga mayor RID.
- Si ningún "router" ha declarado ser el DR entonces el BDR se convierte en el DR.

El proceso real es mucho más complejo, debido a que los mensajes Hello transmitidos incluyen los cambios en los campos grabados en otros "routers", y estos cambios causan eventos en los "routers" que a su vez podrán provocar nuevos cambio u otras acciones. La intención que se esconde tras este mecanismo es doble:

- Que cuando un "router" se active, no debería usurpar la posición del BDR actual aunque tenga un RP superior.
- Que la promoción de un BDR a DR debería ser ordenada y requerir que el BDR acepte sus responsabilidades.

El algoritmo no siempre da lugar a que el "router" de mayor prioridad sea el DR, ni tampoco que el segundo de mayor prioridad sea el BDR.

El DR tiene las siguiente responsabilidades:

- El DR genera para la red los anuncios de los estados de los enlaces, que *inundan* el área y describen esta red a todos los "routers" de todas las redes del área.
- El DR se hace adyacente a otros "routers" de la red. Estas adyacencias son centrales con respecto al proceso de inundación usado para asegurar que los anuncios alcanzan a todos los "routers" del área y que por tanto la base de datos topológica que usan todos permanece igual.

El BDR tiene la siguiente responsabilidad:

- El BDR se hace adyacente a todos los demás "routers" de la red. Esto asegura que cuando ocupe el puesto del DR lo pueda hacer rápidamente.

Formando adyacencias

Después de que se ha descubierto un vecino, asegurado la comunicación bidireccional, y(en una red multiacceso) elegido un DR, se toma la decisión de si se debería formar una adyacencia con uno de sus vecinos:

- En redes multiacceso, todos los "routers" se hacen adyacentes al DR y al BDR.
- En enlaces punto a punto(virtuales), cada "router" forma siempre una adyacencia con el "router" del otro extremo.

Si se toma la decisión de no formar una adyacencia, el estado de la comunicación con el vecino permanece en el estado "2-way".

Las adyacencias se establecen usando paquetes DD("*Database Description"),* que contienen un resumen de la base de datos de estados de enlaces

del emisor. Se pueden usar múltiples paquetes para describir la base de datos: con este fin se emplea un procedimiento de sondeo-respuesta. El "router" con mayor ID se convertirá en maestro, el otro en esclavo. Los paquetes DD enviados por el maestro(sondeos o polls) serán reconocidos por los DDs del esclavo(respuestas). El paquete contiene números de secuencia para asegurar la correspondencia entre sondeos y respuestas. Este proceso se denomina DEP("*Database Exchange Process")*.

El formato del paquete DD se muestra en Figura - Paquete DD de OSPF.

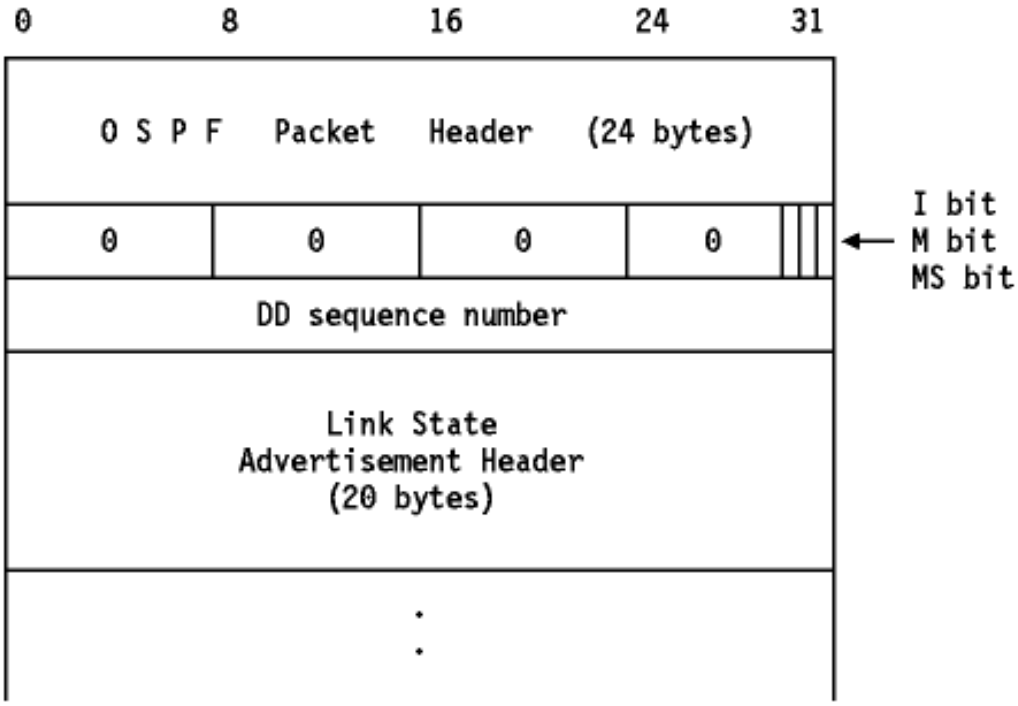


Figura: Paquete DD de OSPF

0
: Reservado, debe ser 0.

I bit
: Bit "Init" o de inicio. Puesto a 1 cuando el paquete es el primero de la secuencia.

M bit
: Bit "More". Indica que siguen más paquetes DD.

MS bit
: Bit maestro/esclavo("Master/slave"). Puesto a 1 cuando el "router" es el maestro, a 0 cuando es el esclavo.

DD sequence number
: Usado para secuenciar la serie de paquetes DD.

El resto del paquete contiene una lista de algunos o todos los contenidos de la base de datos topológica. Cada item de la base de datos es un anuncio de estado del enlace. Los paquetes DD contienen las cabeceras de estos anuncios, que son suficientes para identificar cada anuncio. Esta información se usa en la posterior sincronización de la base de datos. El formato de esta cabecera se muestra en Figura - Cabecera del anuncio del estado del enlace de OSPF.

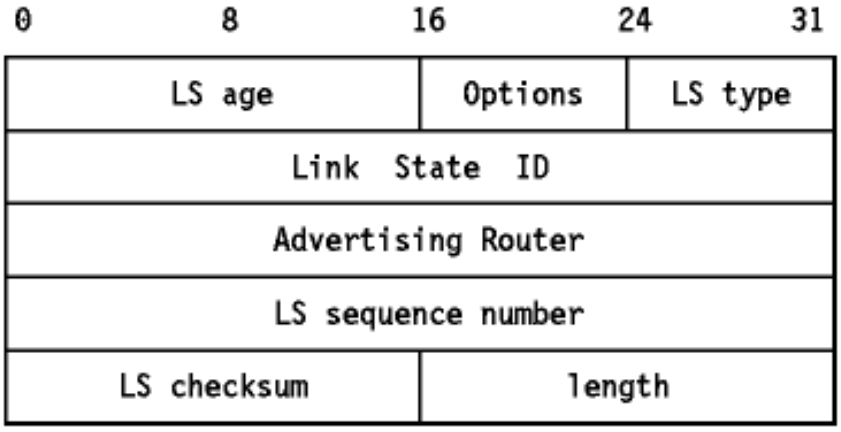


Figura: Cabecera del anuncio del estado del enlace de OSPF

Los campos de la cabecera son:

LS age
: Un número de 16 bits que indica el tiempo en segundos desde el origen del anuncio. En cada salto viaja como parte del procedimiento de inundación. Cuando alcanza su valor máximo, deja de ser usado para determinar las tablas de encaminamiento y se desecha hasta que se

necesite para la sincronización de la base de datos. También se emplea para determinar cuál de dos copias idénticas de un anuncio debería usar un "router".

Options
: Dos bits que describen capacidades opcionales de OSPF. El bit E indica un servicio de encaminamiento externo; se pone a 1 a menos que el anuncio sea para un "router", enlaces de red o un SL("summary link") en un SA("stub area"). El bit E describe rutas para tipos de servicio en adición a TOS 0.

LS type
: Los tipos de anuncio de estado del enlace son:

1
: RL("Router links"). Describen el estado de las interfaces del "router".

2
: NL("Network links"). Describen los "routers" conectados a la red.

3
: SL("Summary links"). Describen "routers" inter-area, intra-AS. Los crean los ABR y permiten que las rutas a redes dentro del AS pero fuera del área sean descritas con precisión.

4
: $$$SL("Summary links"). Describen rutas a la frontera del AS(es decir, a "routers" fronterizos del AS). Los crean los BR. Son muy similares a los de tipo 3.

5
: ASEL("AS external links"). Describen rutas a redes fuera del AS. Los crean los ASBR. Es posible describir una ruta por defecto para el AS de esta forma.

LSD("Link State ID")
: Un ID único para el anuncio que es dependiente de su tipo. Para los tipos 1 y 4es el RID, para los 3 y 5 es un número IP de red, y para el tipo 2 es la dirección IP del DR.

AR("Advertising Router")
: El RID del "router" que originó el anuncio. Para el tipo 1, este campo es idéntico al LSD. Para el 2, es el RID del DR. Para los tipos 3 y 4 es el RID de un ABR. Para el 5, es el RID de un ASBR.

LSN("LS sequence number")
: Usado para permitir la detección de anuncios viejos o duplicados.

LS checksum
: Checksum de todo el anuncio menos el campo LSA("LS age").

Sincronización de las bases de datos

Después de que terminar el DEP("Database Exchange Process"), cada "router" tiene una lista de aquellos anuncios para los que el vecino tiene más instancias actualizadas, que se solicitan por medio de paquetes LSR(*"Link State Request").* La respuesta a un LSR es un LSU("Link State Update") que contiene algunos o todos los anuncios solicitados. Si no se recibe respuesta, se repite la solicitud.

Los anuncios vienen en cinco formatos el formato de un RLA("Router Links Advertisement"; tipo 1) se muestra en Figura - RLA de OSPF.

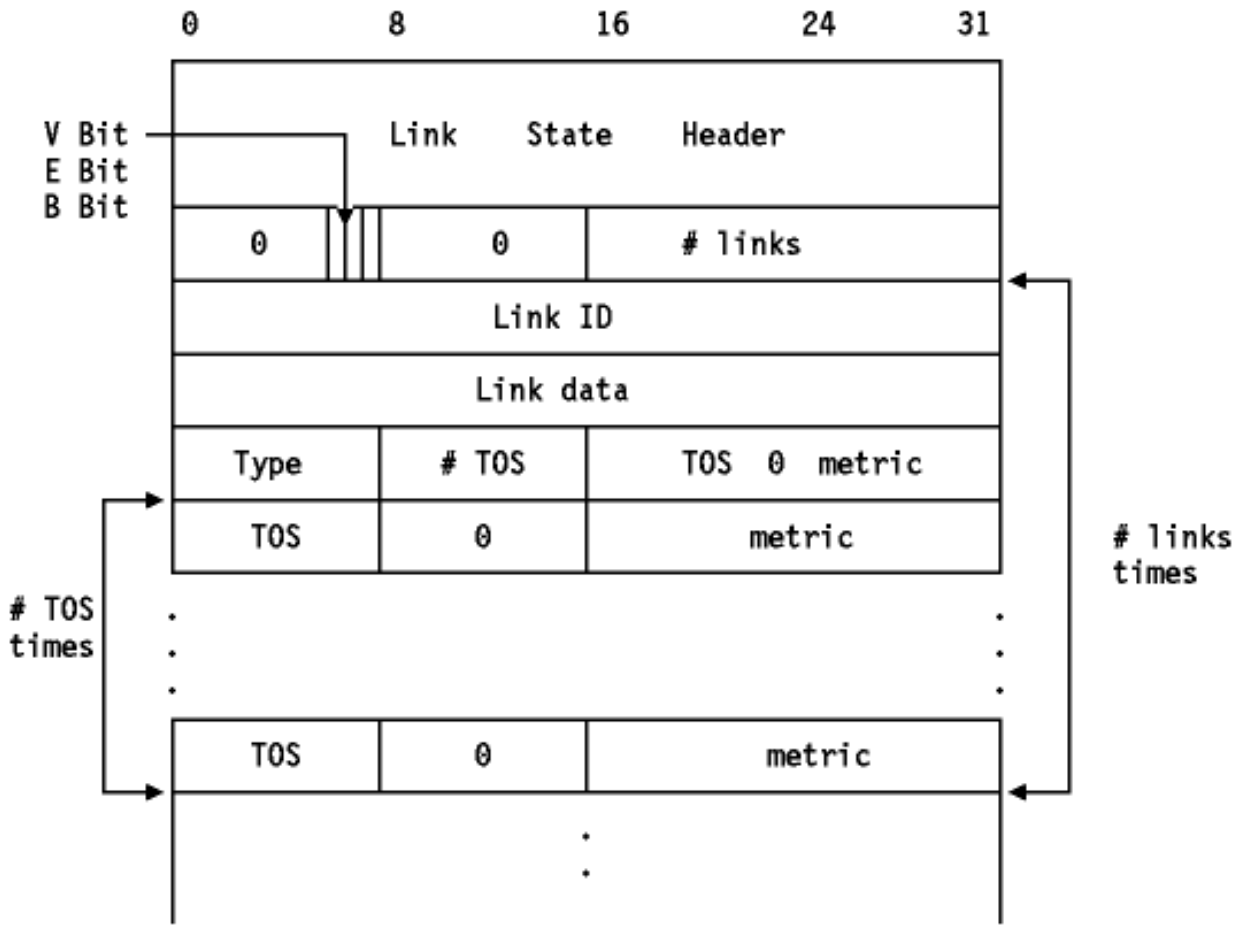


Figura: RLA de OSPF - Este anuncio va encapsulado en un paquete OSPF.

V Bit
: Cuando está a uno, el "router" es el extremo del enlace virtual que usa este área como área de tránsito.

E Bit
: Cuando está a uno, el "router" es un ASBR.

B Bit
: Cuando está a uno, el "router" es un ABR.

links
: El número de enlaces que describe el anuncio.

Link ID
: Identifica el objeto al que conecta este enlace. El valor depende del campo "type"(ver abajo).

1
: RID del vecino

2
: Dir IP del DR

3
: Este valor depende de la ruta inter-area:
- ❏ Para una SN("stub network") es el número IP de red/subred
- ❏ Para un host, es X'FFFFFFFF'
- ❏ Para la ruta externa por defecto el AS es X'00000000'

4
: RIDD del vecino

Link Data
: Este valor también depende del campo "type"(ver el RFC 1583 para más detalles).

Type
: A qué conecta el enlace.

1
: Conexión punto a punto con otro "router"

2
: Conexión con una red de tránsito

3
: Conexión a una SN o a un host

4
: Virtual link

metric
: El número de diferentes métricas de TOS para este enlace además de la métrica para TOS 0.

TOS 0 metric
: El coste de usar el enlace para TOS 0. Todos los paquetes de encaminamiento de OSPF se envían con el campo IP TOS a cero.

TOS
: Tipo IP de servicio al que se refiere la métrica. El RFC 1349 define los posibles valores TOS en una cabecera IP(ver IP("Internet Protocol")) con una secuencia de 4 bits. OSPF codifica estos códigos tratando la secuencia como un número y duplicándolo(a continuación del campo TOS hay un bit reservado puesto a cero, que OSPF guarda para su posible inclusión en el futuro en el valor TOS. Hay cinco valores definidos:

Tabla: Valores de "Type of Service"

El coste de usar este "router" para el tráfico del tipo de servicio especificado por el campo "Type of Service".

Como ejemplo, suponer el enlace punto a punto entre los "routers" RT1 (dirección IP: 192.1.2.3) y RT6 (dirección IP: 6.5.4.3) es como un enlace satélite. Para favorecer el uso de esta línea para el tráfico banda alta, el administrador AS puede fijar artificialmente una métrica baja para ese TOS. RT1 originaría los siguientes anuncios de estado del enlaces(suponiendo que RT1 es un ABR y no un ASBR):

```
; RT1's router links advertisement

LS age = 0                      ; always true on origination
LS type = 1                     ; indicates router links
Link State ID = 192.1.2.3       ; RT1's Router ID
Advertising Router = 192.1.2.3  ; RT1
bit E = 0                       ; not an AS boundary router
bit B = 1                       ; area border router
#links = 1
       Link ID = 6.5.4.3        ; neighbor router's Router ID
       Link Data = 0.0.0.0      ; interface to unnumbered SL
       Type = 1                 ; connects to router
       # other metrics = 1
       TOS 0 metric = 8
           TOS = 2              ; high bandwidth
           metric = 1           ; traffic preferred
```

El formato de un NLA("Network Links Advertisement"; tipo 2) se muestra en Figura - NLA de OSPF.

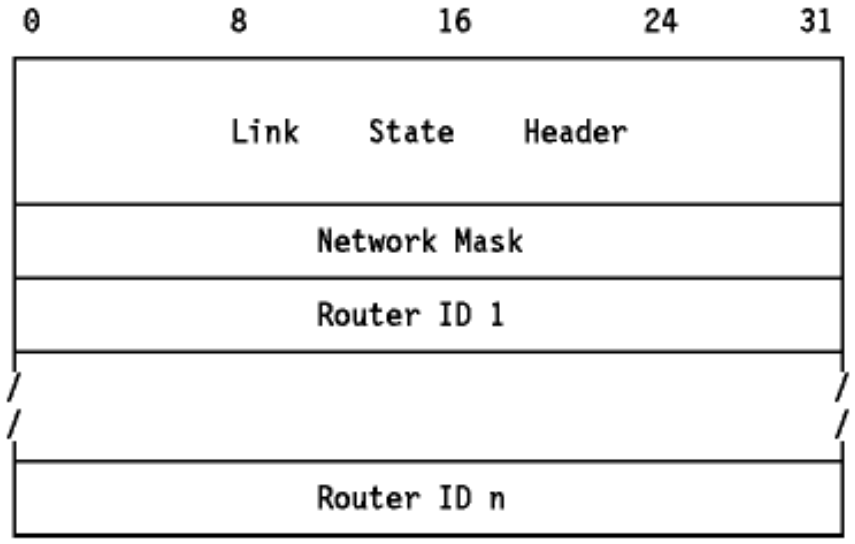


Figura: NLA de OSPF - Este anuncio va encapsulado en un paquete OSPF.

Network Mask
: La máscara IP para la red. Por ejemplo, una red de clase C tendría una máscara de 255.0.0.0.

Router ID 1-n
: Las direcciones IP de todos los "routers" de la red que sean adyacentes al DR (incluyendo al "router" emisor). El número de "routers" en una lista se deduce del campo "lenght" de la cabecera.

El formato un SLA("Summary Links Advertisement"; tipo 3 o 4) se muestra en Figura - SLA de OSPF.

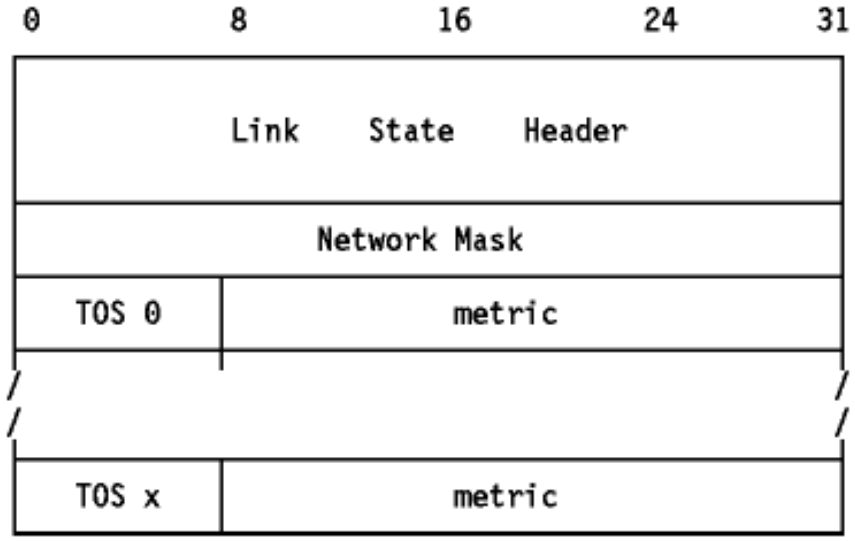


Figura: SLA de OSPF - Este anuncio va encapsulado en un paquete OSPF.

Network Mask
: Para un anuncio de tipo 3, es la máscara IP para la red. Para uno de tipo 4 no tiene significado y debe ser cero.

TOS 0
: zero

metric
: El coste de esta ruta para este tipo de servicio en las mismas unidades usadas para las métricas TOS en los anuncios tipo 1.

TOS x
: Cero o más entradas para tipos adicionales de servicio. El número de entradas se puede determinar a partir del campo "length" de la cabecera.

El formato de una ELA("External Links Advertisement") se muestra en Figura - ELA de OSPF.

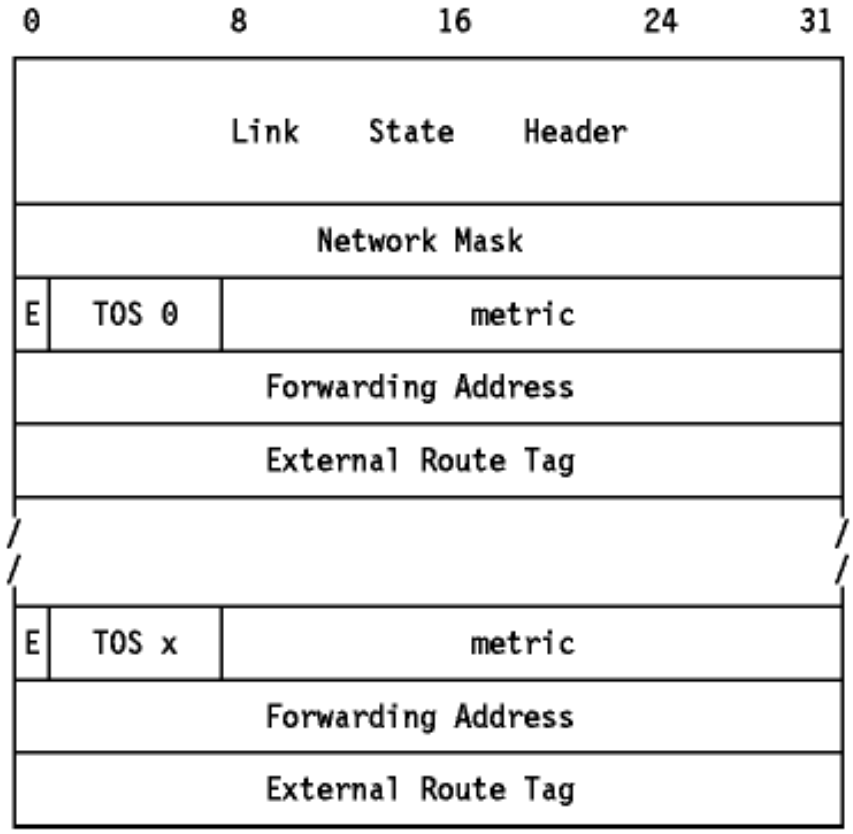


Figura: ELA de OSPF - Este anuncio va encapsulado en un paquete OSPF.

Network Mask
: La máscara IP para la red.

Bit E
: El tipo de métrica externa. Si está a uno es 2, sin no es1.

1
: La métrica se puede comparar directamente con las métricas del estado del enlace de OSPF

2
: La métrica se considera mayor que todas las métricas del estado del enlace de OSPF

TOS 0
: zero

metric
: El coste de esta ruta. La interpretación depende del E-bit.

Forwarding Address
: La dirección IP a la que se ha de dirigir el tráfico de datos del tipo de servicio especificado para ese destino. El valor cero indica que el tráfico se debería enviar al ASBR que originó el anuncio.

External Route Tag
: Un valor de 32 bits conectado a la ruta externa por un ASBR. OSPF no define o usa este valor, pero Interacción de BGP con OSPF define su uso cuando se emplea BGP como protocolo de encaminamiento externo.

TOS x
: Cero o más entradas para tipos adicionales de servicio. El número de entrada se puede determinar a partir de la longitud del campo "length" de la cabecera.

Cuando se han respondido los paquetes LSR, las bases de datos se sincronizan y los "routers" se describen como totalmente adyacentes. La adyacencia se añade a los anuncios de los dos "routers" correspondientes.

Calculando la tabla de encaminamiento

Usando como entrada las bases de datos de estados de enlaces de las áreas con las que está conectado, un "router" ejecuta el algoritmo SPF para construir su tabla de encaminamiento. La tabla de encaminamiento siempre se construye es de cero: nunca se hacen actualizaciones a una tabla ya existente. Una tabla de encaminamiento vieja no se desecha hasta que se han identificado los cambios entre las dos tablas. Brevemente, el cálculo consiste en los pasos indicados abajo. Ver el RFC 1583 para más detalles sobre la implementación del algoritmo.

1. Las rutas intra-area se calculan construyendo el árbol mínimo para cada área conectada usando el mismo "router" como raíz del árbol. El "router" calcula además si el área puede actuar como área de tránsito para enlaces virtuales.
2. Las rutas inter-area se calculan examinando los SLA. Para los ABR(que forman parte de la troncal) sólo se utilizan los anuncios correspondientes a la troncal(es decir, un ABR siempre encaminará tráfico inter-area a través de la troncal).
3. Si el "router" está conectado a una o más áreas de tránsito, el "router" sustituye las rutas que haya calculado por rutas que pasen por áreas de tránsito si estas son mejores.
4. Las rutas externas se calculan por examinando los anuncios externos del AS. Las localizaciones de los ASBR ya se conocen debido a que se determinan como cualquier otra ruta intra-area o inter-area.

Cuando el algoritmo produce rutas de igual coste, OSPF puede balancear uniformemente la carga a través de ellas. El número máximo de rutas iguales admitidas depende de la implementación.

Anunciando los estados de los enlaces

Un "router" anuncia periódicamente el estado de su enlace, por lo que la ausencia de un anuncio reciente indica a los vecinos del "router" que no está activo. Todos los "routers" que hayan establecido comunicación bidireccional con un vecino ejecutan un contador de inactividad para detectar ese suceso. Si no se resetea el contador, al final se desbordará y el evento asociado sitúa el estado del vecino en "down". Esto significa que la comunicación se debe establecer desde cero, incluyendo la resincronización de las bases de datos. Un "router" también relanza sus anuncios cuando su estado cambia.

Un "router" puede lanzar diversos anuncios para cada área. Estos se propagan a través del área por el procedimiento de inundación. Cada "router" emite un RLA. Si el "router" es además el DR para una o más de las redes del área, originará NLAs para estas. Los ABR generan una SLA para cada destino inter-area conocido. Los ASBR originan un ASL para cada destino externo conocido. Los destinos se anuncian uno cada vez de tal forma que el cambio de una sola ruta puede inundar la red sin tener que enviar el resto de las rutas. Durante el proceso de inundación, un sólo LSU puede llevar muchos anuncios.

3.3.4.2 Resumen de características de OSPF

OSPF es un protocolo de encaminamiento complejo, como las anteriores secciones han hecho patente. Los beneficios de esta complejidad(sobre RIP) son los siguientes:

- Debido a las bases de datos de estados de enlaces sincronizadas, los "router" OSPF convergerán mucho más rápido que los "routers" RIP tras cambios de topología. Este efecto se hace más pronunciado al aumentar el tamaño del AS.
- Incluye encaminamiento TOS("*Type of Service")* diseñado para calcular rutas separadas para cada tipo de servicio. Para cada destino, pueden existir múltiples rutas, cada una para uno o más TOSs.
- Utiliza métricas ponderadas para distintas velocidades el enlace. Por ejemplo, un enlace T1 a.544 Mbps podría tener una métrica de 1 y un SLP a 9600 bps una de 10.
- Proporciona balanceamiento de la carga ya que una pasarela OSPF puede emplear varios caminos de igual coste mínimo.
- A cada ruta se le asocia una máscara de subred, permitiendo subnetting de longitud variable(ver Subredes) y supernetting (ver CIDR(Classless Inter-Domain Routing)).
- Todos los intercambios entre "routers" se pueden autentificar mediante el uso de passwords.
- OSPF soporta rutas específicas de hosts, redes y subredes.
- OSPF permite que las redes y los hosts contiguos se agrupen juntos en *áreas* dentro de un AS, simplificando la topología y reduciendo la cantidad de información de encaminamiento que se debe intercambiar. La topología de un área es desconocida para el resto de las áreas.
- Minimiza los broadcast permitiendo una topología de grafo más compleja en la que las redes multiacceso tienen un DR que es responsable de describir esa red a las demás redes del área.
- Permite el intercambio de información de encaminamiento externa, es decir, información de encaminamiento obtenida de otro AS.
- Permite configurar el encaminamiento dentro del AS según una topología virtual más que sólo las conexiones físicas. Las áreas se pueden unir usando enlaces virtuales que crucen otras áreas sin requerir encaminamiento complicado.
- Permite el uso de enlaces punto a punto sin direcciones IP, lo que puede ahorrar recursos escaso en el espacio de direcciones IP.

Referencias

Una descripción detallada de IP se puede encontrar en los siguientes RFCs:

- *RFC 1245 - Análisis del protocolo OSPF*
- *RFC 1246 - Experiencia con el protocolo OSPF*
- *RFC 1253 - OSPF Versión 2: MIB("Management Information Base")*
- *RFC 1370 - Condiciones de aplicabilidad de OSPF*
- *RFC 1583 - OSPF Versión 2*

3.3.4.3 IS-IS de OSI("Intermediate System to Intermediate System")

IS-IS es un protocolo similar a OSPF: también emplea el estado del enlace, el algoritmo SPF("Shortest Path First"; ver Estado del enlace, primero el camino más corto para más detalles). Sin embargo, IS-IS es un protocolo OSI usado para los paquetes CLNP(*"Connectionless Network Protocol")* en un dominio de encaminamiento. CLNP es el protocolo OSI más comparable a IP.

El IS-IS integrado extiende IS-IS para compararse a TCP/IP. Se describe en el RFC 1195. Su meta es proporcionar un sólo(y eficiente) protocolo de encaminamiento para TCP/IP y para OSI. Su diseño hace uso del protocolo de encaminamiento OSI IS-IS, aumentado con información IP específica, y proporciona apoyo explícito para el subnetting IP, máscaras de red variables, encaminamiento TOS, y encaminamiento externo, además de recurso para la autentificación. El IS-IS integrado se basa en el mismo algoritmo de encaminamiento que OSPF.

No emplea encapsulación mutua de los paquetes IP y CLNP: ambos tipos se envían tal como son, ni cambia el comportamiento del "router" como ambas pilas de protocolos podrían esperar. Se comporta como un IGP en una red TCP/IP y en una red OSI. El único cambio es la adición de información adicional relacionada con IP.

IS-IS usa el término IS(*"Intermediate System")* para referirse a un "router" IS-IS router, pero usaremos el termino "router", ya que se usa con libertad en el est IS-IS integrado.

IS-IS agrupa las redes en dominios de modo análogo a OSPF. Un *dominio de encaminamiento* es análogo a un AS, y se subdivide en *áreas*,

exactamente como OSPF. Aquí hay una descripción de los aspectos más importantes del encaminamiento IS-IS. Cuando es posible, se hacen comparaciones con conceptos equivalentes de OSPF.

- Los "routers" se dividen en "routers" de nivel 1, que no saben nada de la topología fuera de sus áreas, y de nivel 2, que conocen la topología de nivel superior, pero no saben nada de la topología de dentro de las áreas, a menos que sean también "routers" de nivel 1.
- Un "router" de nivel 1 puede pertenecer a más de un área, pero a diferencia de OSPF esto no se hace con propósitos de encaminamiento sino para facilitar la gestión del dominio, y normalmente por poco tiempo. Un "router" de nivel 1 reconoce a otro como un vecino si están en la misma área.
- Un "router" de nivel 2 reconoce a todos los demás "routers" de nivel 2 como vecinos. Un "router" de nivel 2 puede ser también un "router" de nivel 1 en un área, pero no en más.
- Un "router" de nivel 1 en IS-IS no puede tener un enlace con un "router" externo(en OSPF un "router" interno puede ser una ASBR).
- Hay una troncal de nivel 2 que contiene todos los "routers" de nivel dos, pero a diferencia de OSPF, debe estar conectada físicamente.
- El esquema de dirección OSI identifica explícitamente el área objetivo de un paquete, permitiendo una selección sencilla de las rutas del modo siguiente:
  - Los "routers" de nivel 2 encaminan hacia el área sin importarles su estructura interna.
  - Los "routers" de nivel 1 encaminan hacia el destino si está en su área, o al "router" de nivel 2 más cercano no es así.
- Las redes multiacceso usan el concepto de DR("Designated Router"). Para evitar el problema "$n(n-1)/2$" descrito en OSPF, IS-IS implementa un *pseudonodo* para la LAN. Se considera que cada "router" conectado a la LAN tiene un enlace con el pseudonodo, pero ninguno con los demás "routers" de la LAN. El DR actúa representando al pseudonodo.

IS-IS integrado permite una mezcla considerable de las dos pilas de protocolo, sujeto a ciertas restricciones sobre la topología. Se definen tres tipos de rutas:

IP-only
: Un "router" que usa IS-IS como protocolo de encaminamiento y para IP y no soporta protocolos OSI(por ejemplo, tales "routers" no serían capaces de transmitir paquetes CLNP).

OSI-only
: Un "router" que usa IS-IS como protocolo de encaminamiento para OSI pero no usa IP.

Dual
: Un "router" que usa IS-IS como un único protocolo de encaminamiento integrado tanto para IP como para OSI.

Es posible tener un dominio mixto que contenga "routers" IS-IS, algunos de los cuales son "IP-only", algunos "OSI-only" y algunos del tipo "dual". Cada área dentro de un dominio se configura como OSI, IP o "dual". Las áreas que han de soportar tráfico mixto deben tener todos los "routers" de nivel 1 del tipo "dual". Similarmente, los "routers" de nivel 2 en un dominio mixto deben ser "dual" si el tráfico mixto se tiene que encaminar entre áreas.

3.3.4.4 Co-existencia de los protocolos de encaminamiento de TCP/IP y OSI sin IS-IS

Como sugiere su nombre, el IS-IS integrado ofrece una solución de encaminamiento para redes multi-protocolo. OSPF, como otros protocolos de encaminamiento TCP/IP, utiliza una estrategia llamada SIN("*Ships In the Night")* para manejar cuestiones de coexistencia. En la estrategia SIN, cada "router" multi-protocolo ejecuta un proceso separado para cada capa de red(IP y OSI). Un "router" SIN permite que los gestores de la red inserten nuevos protocolos de encaminamiento basados en SIN, como OSPFS, uno a uno, pero los protocolos existen con independencia unos de otros.

La mayoría de los distribuidores de estos protocolos ha elegido seguir con SIN, debido a la pujanza de TCP/IP. Unos cuantos han anunciado que en el futuro apoyarán el IS-IS integrado.

Tabla de contenidos Protocolos de encaminamiento exterior

3.4 Protocolos de encaminamiento exterior

Los ERPs o EGPs*("Exterior Routing Protocols"* o "*Exterior Gateway Protocols")* se usan para intercambiar información de encaminamiento entre distintos ASs.

Los EGPs más usados son dos

- EGP("Exterior Gateway Protocol", ver EGP("Exterior Gateway Protocol")).
- BGP("Border Gateway Protocol", ver BGP("Border Gateway Protocol")).

BGP está sustituyendo progresivamente a EGP.

3.4.1 EGP("Exterior Gateway Protocol")

EGP es un *protocolo estándar*. Su status es *recomendado*.

EGP es el protocolo utilizado para el intercambio de información de encaminamiento entre pasarelas *exteriores*(que no pertenezcan al mismo AS).

Las pasarelas EGP sólo pueden retransmitir información de accesibilidad para las redes de su AS. La pasarela debe recoger esta información, habitualmente por medio de un IGP, usado para intercambiar información entre pasarelas del mismo AS(ver Figura - La troncal ARPANET).

EGP se basa en el sondeo periódico empleando intercambios de mensajes *Hello/I Hear You*, para monitorizar la accesibilidad de los vecinos y para sondear si hay solicitudes de actualización. EGP restringe las pasarelas exteriores al permitirles anunciar sólo las redes de destino accesibles en el AS de la pasarela. De esta forma, una pasarela exterior que usa EGP pasa información a sus vecinos EGP pero no anuncia la información de accesibilidad de estos(las pasarelas son vecinos si intercambian información de encaminamiento) fuera del AS. Tiene tres características principales:

- Soporta un protocolo NAP("*Neighbor Acquisition Protocol)*. Dos pasarelas se pueden considerar vecinas si están conectadas por una red que es transparente para ambas. EGP no especifica la forma en que una pasarela decide inicialmente que quiere ser vecina de otra. Para convertirse en vecina, debe enviar un mensaje *"Acquisition confirm"* como respuesta a un *Acquisition Request*. Este paso es necesario para obtener información de encaminamiento de otra pasarela.
- Soporta un protocolo NR("*Neighbor Reachability")*. La pasarela lo usa para mantener información en tiempo real sobre la accesibilidad de sus vecinos. El protocolo EGP proporciona dos tipos de mensajes para ese fin: un mensaje *Hello* y un mensaje *I Hear You*(respuesta a Hello).
- Soporta mensajes de actualización(o mensajes NR) que llevan información de encaminamiento. No se requiere ninguna pasarela para enviar mensajes NR a otra pasarela, excepto como respuesta a una petición de sondeo("*poll request")*.

Para realizar estas tres funciones básicas, EGP define 10 tipos de mensajes:

Acquisition Request
: Solicita que una pasarela se convierta en vecina

Acquisition Confirm
: Respuesta afirmativa a un "acquisition request"

Acquisition Refuse
: Respuesta negativa a un "acquisition request"

Cease Request
: Solicitud de terminación de la relación de vecindad

Cease Confirm
: Confirmación para que cesen las peticiones

Hello
: Solicitud de respuesta e un vecino, si está vivo

I Hear You
: Respuesta el mensaje Hello

Poll Request
: Solicitud de la tabla de encaminamiento de la red

Routing Update
: Información de accesibilidad de la red

Error
: Respuesta a un mensaje incorrecto

Consideremos el mensaje de actualización mostrado en Figura - mensaje de actualización EGP.

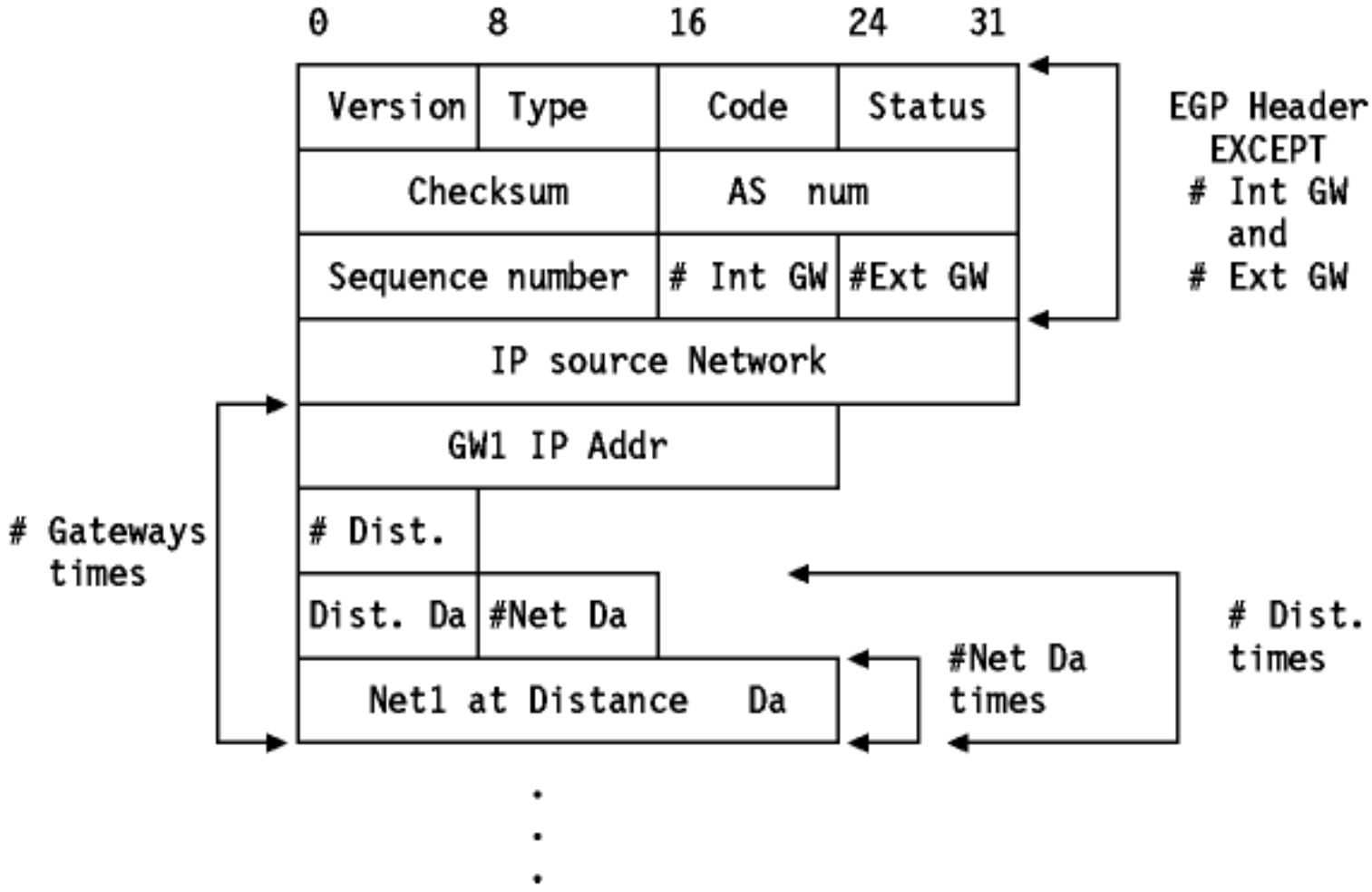


Figura: mensaje de actualización EGP

Los distintos campos son los siguientes(no se considera la cabecera EGP; referirse al RC 904 para más detalles):

#Int GW
: Número de pasarelas interiores que aparecen en el mensaje.

#Ext GW
: Número de pasarelas exteriores que aparecen en el mensaje.

IP Source Network
: La dirección IP de red para la que se mide la accesibilidad.

GW1 IP addr
: Dirección IP sin el número de red de la pasarela para la que se miden las distancias.

#Dist.
: Número de distancias en el bloque de la pasarela.

Dist.Da
: Valor de la distancia.

#Net Da
: Número de redes a una distancia dada(Da).

Net1 at distance Da
: Número IP de la red accesible por GW1 a una distancia Da de GW1.

Como se indicó arriba, los mensajes EGP asocian un descriptor *"distances"* a cada ruta. Pero EGP *no interpreta estos valores.* Simplemente sirven como indicación de la accesibilidad o inaccesibilidad de una red(un valor de 255 significa que la red es

inalcanzable). El valor no se puede usar para calcular cuál es la más corta de dos rutas a menos que ambas pertenezcan al mismo AS. Por esta razón, EGP no se puede usar como algoritmo de encaminamiento. Como resultado sólo habrá una única ruta del exterior de la pasarela a una red.

3.4.2 BGP("Border Gateway Protocol")

Nota: Hay cuatro versiones de BGP. Cuando se dice BGP, suele hacerse referencia a la versión 3, a menos que el documento sea anterior a esta versión. Esta sección describe BGP 3 y BGP-4 en BGP("Border Gateway Protocol" ,Versión 4). BGP-1 y BGP-2, descritos en los RFCs 1105 y RFC 1163, son obsoletos. Los cambios desde BGP-1 y BGP-2 a BGP-3 se documentan en el apéndice 2 y el 3 del RFC 1267.

3.4.2.1 BGP 3("Border Gateway Protocol", Versión 3)

BGP-3 es un *borrador*. Su status es *electivo*. Se describe en el RFC 1267.

BGP-3 es un protocolo de encaminamiento *inter-AS* basado en la experiencia obtenida de EGP(ver EGP("Exterior Gateway Protocol")). A diferencia de otros protocolos de encaminamiento que se comunican mediante paquetes o datos, BGP-3 está orientado a conexión; utiliza TCP como protocolo de transporte . El número de puerto bien conocido es el 179. Ver TCP("Transmission Control Protocol") para información sobre TCP y los números de puerto.

Recuérdese que EGP se diseñó como un protocolo para intercambiar información de encaminamiento entre ASs, más que como un verdadero protocolo de encaminamiento. Debido a que la información de encaminamiento inter-AS no está disponible, EGP no puede detectar la presencia de un bucle causado por un conjunto de "routers" que creen que uno de ellos puede alcanzar otro AS al que ninguno de ellos está conectado. Un problema adicional con EGP tiene que ver con la cantidad de información intercambiada; a medida que el número de redes IP que conoce NSFNET aumenta, el tamaño de los mensajes NR aumenta también y la cantidad de tiempo necesaria para procesarlos se hace significativa.

BGP-3 ha sustituido a EGP en la troncal NSFNET por estas razones. Si embargo, BGP-3, tal como lo describe el RFC 1268, no requiere que NSFNET o cualquier otra troncal juegue un papel central, en comparación con el carácter de núcleo que jugó ARPANET en los primeros tiempos de Internet. En vez de eso, BGP-3 ve Internet como una colección de Ass, y no tiene en cuenta la topología interna de un AS ni el IGP o IGPs que utilice.

Antes de describir BGP-3, definiremos algunos términos usados en BGP-3:

BGP speaker(BGPS)
: Un sistema que ejecuta BGP.

BGP neighbors
: Un par de BGPSs intercambiando información de encaminamiento inter-AS. Lo vecinos BGP pueden ser e dos tipos:

Internal
: Un par de BGPSs en el mismo AS. Deben presentar a los vecinos externos al AS una imagen consistente de su propio AS.

External
: Un par de BGPSs en distintos Ass. Los vecinos externos("external") deben estar conectados por una conexión BGP como se define más abajo. Esta restricción significa que en la mayoría de los casos en los que un AS tenga múltiples conexiones inter-AS de tipo BGP, también requerirá múltiples BGPSs.

BGP session
: Una sesión BGP entre vecinos BGP que se intercambian información de encaminamiento por medio de BGP. Los vecinos monitorizan el estado de la conexión enviando un mensajes *"keepalive"* regularmente(el intervalo recomendado es de 30 segundos(9).

AS Border Router (ASBR)
: Un "router" con conexión a múltiples ASs.
Nota: La nomenclatura para este tipo de "router" es algo variada. Además, OSPF también usa el término ASBR. Usaremos ASBR para BGP-3, si bien BGP-3 define dos tipos de ASBR, dependiendo de su relación topológica con el BGPS al que se refieren.

internal
: Un "router" a un salto de distancia en mismo AS que el BGPS.

external
: Un "router" a un salto de distancia en distinto AS que el BGPS.

La dirección IP de un ASBR se especifica como el siguiente salto cuando BGP-3 anuncia una ruta AS(ver abajo) a uno de sus vecinos. Dicho ASBR debe compartir una conexión física con los BGPSs emisor y receptor. Si un BGPS detecta un ASBR como

siguiente salto, el BGPS debe conocerlo previamente por sus sondeos.

AS connection
: BGP-3 define dos tipos de conexión inter-AS.

physical connection
: Un AS comparte una red física con otro AS, y esta red está conectada a al menos un ASBR de cada AS. Como estos dos ASBRs comparten una red, se pueden enviar paquetes sin requerir ningún protocolo de encaminamiento inter-AS o intra-AS(es decir, no requieren de IGP ni de EGP para comunicarse).

BGP connection
: Una conexión BGP significa que hay una sesión BGP entre un par de BGPSs, uno en cada AS, y esta sesión se usa para comunicar las rutas a través de los ASBRs que se pueden emplear para redes específicas. BGP-3 requiere que los BGPSs estén en la misma red al igual que los ASBRs conectados físicamente, por lo que la sesión BGP es independiente de todos los protocolos inter-AS o intra-AS. Los BGPSs no han ser ASBRs, y viceversa. De hecho, los BGPSs no han de ser siquiera "routers": para un host es bastante fácil proporcionar la función BGP y pasar información de encaminamiento exterior a uno o más ASBRs con otro protocolo.
: **Nota:** El término conexión BGP se puede usar para referirse a una sesión entre dos BGPSs en el mismo AS.

Traffic type
: BGP-3 categoriza el tráfico en un AS en dos tipos:

local
: El tráfico que se origina o que termina en ese AS. Es decir, o bien la dirección fuente o bien la de destino están en el AS.

transit
: Tráfico no local.

Uno de los objetivos de BGP es minimizar este tipo de tráfico.

AS Type
: Un AS se clasifica en uno de tres tipos:

stub
: Un SAS("stub AS") tiene una sola conexión inter-AS con otro AS, y solo lleva tráfico de tipo "local".

multihomed
: Un AS "multihomed"(multipuerto) tiene conexiones a uno o más ASs pero rechaza llevar tráfico de tipo "transit".

transit
: Un AS de tipo "transit" tiene conexiones a uno o más Ass y lleva tráfico de tipo "transit". El AS puede imponer restricciones en el tráfico que llevará.

AS number
: Un número de 16 bits que identifica unívocamente el AS. Es el mismo número que usan GGP y EGP.

AS path
: Una lista de todos los números AS que atraviesa una ruta al intercambiar información de encaminamiento. Más que intercambiar simples valores de métrica, BGP-3 comunica rutas enteras a sus vecinos.

Routing Policy
: Un conjunto de reglas que constriñen el encaminamiento para adecuarse a los deseos de la autoridad que administra el AS. Las políticas de encaminamiento no están definidas en el protocolo BGP-3, pero están seleccionadas por la autoridad AS y se presentan a BGP-3 en forma de datos de configuración específicos de la implementación. Las políticas de encaminamiento las puede seleccionar la autoridad del AS del modo que considere oportuno. Por ejemplo:

- Un AS "multihomed" puede rechazar actuar como AS "transit". Lo consigue no anunciándose a otras redes más que a las conectadas directamente con él.
- Un AS "multihomed" puede limitarse a ser de tipo "transit" para un número restringido de ASs adyacentes. Lo consigue anunciando su información de encaminamiento sólo a este conjunto.
- Un AS puede seleccionar que AS externo se debería usar para llevar tráfico de tipo "transit". También puede aplicar criterios de rendimiento al seleccionar las rutas al exterior:
- Un AS puede optimizar el tráfico para usar rutas cortas.
- Un AS puede seleccionar rutas de tránsito según la calidad el servicio en los saltos intermedios. Esta calidad se podría determinar con mecanismos ajenos a BGP-3.

De la definición anterior se puede ver que un SAS o un AS "multihomed" tienen las mismas propiedades topológicas que en la arquitectura ARPANET: nunca actúan como AS intermedios("intermediate AS") en una ruta inter-AS. En la arquitectura ARPANET, EGP bastaba para que esta clase de ASs intercambiase información de accesibilidad con sus vecinos, y esto sigue siendo cierto con BGP-3. Por tanto, un SAS o un AS "multihomed" pueden seguir usando EGP(o cualquier otro protocolo adecuado) para operar con un AS de tipo "transit". Sin embargo, el RFC 1268 recomienda usar BGP. Adicionalmente, en un AS "multihomed", es probable que BGP proporcione un encaminamiento inter-AS más óptimo que EGP, ya que EGP no considera la distancia.

Selección de la ruta

Cada BGPS

Cada BGPS debe evaluar distintas rutas a un destino desde los ASBRs de la conexión con el AS, seleccionar la que mejor cumpla la política de encaminamiento y luego anunciar esa ruta a todos sus vecinos en la conexión con el AS.

BGP-3 es un protocolo vector-distancia pero, a diferencia de los protocolos vector-distancia tradicionales tales como RIP, en los que existe sólo una métrica, BGP determina un orden de preferencia al aplicar una función que mapea cada ruta a un valor de prioridad y selecciona la ruta que tanga el mayor valor. Esta función la genera la implementación de BGP-3 según la información de configuración.

Cuando hay múltiples rutas hasta un destino, BGP-3 las mantiene todas pero sólo anuncia la de mayor preferencia. Esta estrategia permite cambiar rápidamente a una ruta alternativa cuando falla la principal.

Políticas de encaminamiento

El RFC 1268 incluye un conjunto recomendado de políticas de encaminamiento para todas las implementaciones:

- Una implementación de BGP-3 debería ser capaz de controlar las rutas que anuncia. La granularidad de este control debería estar al menos al nivel de red para las rutas anunciadas y al del AS para los receptores.
- BGP-3 debería permitir una política de ponderación para las rutas. A cada AS se le puede asignar un peso específico de modo que la ruta preferida a un destino es la de menor peso resultante de la agregación de los pesos de los ASs.
- BGP-3 debería permitir una política de exclusión de un AS de todas las posibles rutas . Esto se puede hacer con una variante de la política anterior; a cada AS a excluir se le da un peso "infinito" y el proceso de selección de rutas se encargará de rechazar las rutas de peso infinito.

Consistencia de un AS

BGP-3 requiere que un AS tipo "transit" presente el mismo aspecto a todo AS que emplee sus servicios. Si el AS tiene múltiples BGPSs, deben estar de acuerdo sobre dos aspectos de la topología: intra-AS e inter-AS. Como BGP no maneja el encaminamiento intra-AS en absoluto, el protocolo de encaminamiento interior debe dar una visión consistente de la topología intra-AS. Naturalmente, un protocolo tal como OSPF(ver OSPF("Open Shortest Path First Protocol") Versión 2) o IS-IS Integrado (ver IS-IS de OSI("Intermediate System to Intermediate System")) que implementa la sincronización de bases de datos de "routers" se presta por sí misma a este papel. La consistencia de la topología eterna la *pueden* proporcionar todos los BGPSs del AS que tengan sesiones entre sí, pero BGP-3 no pide que se utilice este método, sólo que se mantenga la consistencia.

Intercambio de información de encaminamiento

BGP-3 sólo anuncia las rutas usadas con sus vecinos. Es decir, se adapta al paradigma habitual salto-a-salto de Internet, incluso si tiene información adicional en la forma de rutas AS y aunque fuese capaz de informar a un vecino de una ruta que él mismo no usa.

Cuando dos BGPSs forma una sesión BGP, comienzan a intercambiar todas sus tablas de encaminamiento. La información de encaminamiento se intercambia por medio de mensajes UPDATE(ver abajo). Como la información de encaminamiento contiene la ruta completa para cada destino en forma de una lista de números de AS además de la información normal de accesibilidad y del siguiente salto empleadas en protocolos vector-distancia, se puede usar para eliminar los bucles y para eliminar el *problema de la cuenta hasta infinito* de RIP. Después de que los vecinos han efectuado su intercambio inicial de sus bases de datos, sólo se envían actualizaciones de esa información.

Formato de mensaje de lBGP-3

Todos los mensajes BGP-3 tienen en común un formato básico. Varían en longitud de 19 a 4096 bytes, se transmiten sobre TCP y se procesan en su totalidad(no se procesan hasta que se han recibido por completo). Cada mensaje tiene una cabecera mostrada en Figura - Cabecera BGP-3.

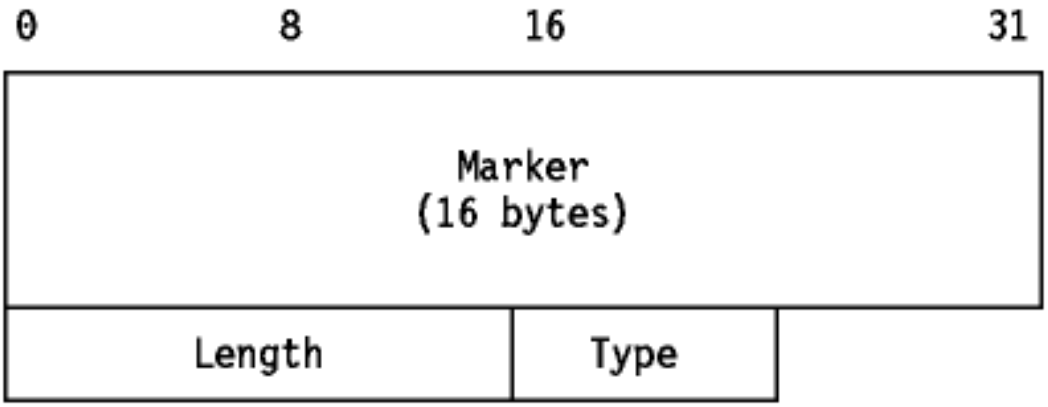


Figura: Cabecera BGP-3

Marker
: Un valor que el receptor puede predecir, usado para la autentificación y para identificar pérdidas de sincronización. Se rellena con unos cuando " Authentication Code" es 0(ver abajo).

Length
: Longitud total del mensaje, incluyendo la cabecera, en bytes. El mensaje no se puede rellenar o engordar ya que en muchos casos la longitud se utiliza para calcular la longitud del último campo del mensaje.

type
: Un valor sin signo de 8 bits.

1
: OPEN (10)

2
: UPDATE

3
: NOTIFICATION

4
: KEEPALIVE

Los mensajes OPEN se usan para iniciar la sesión BGP-3. El formato se muestra en Figura - Mensaje OPEN de BGP-3.

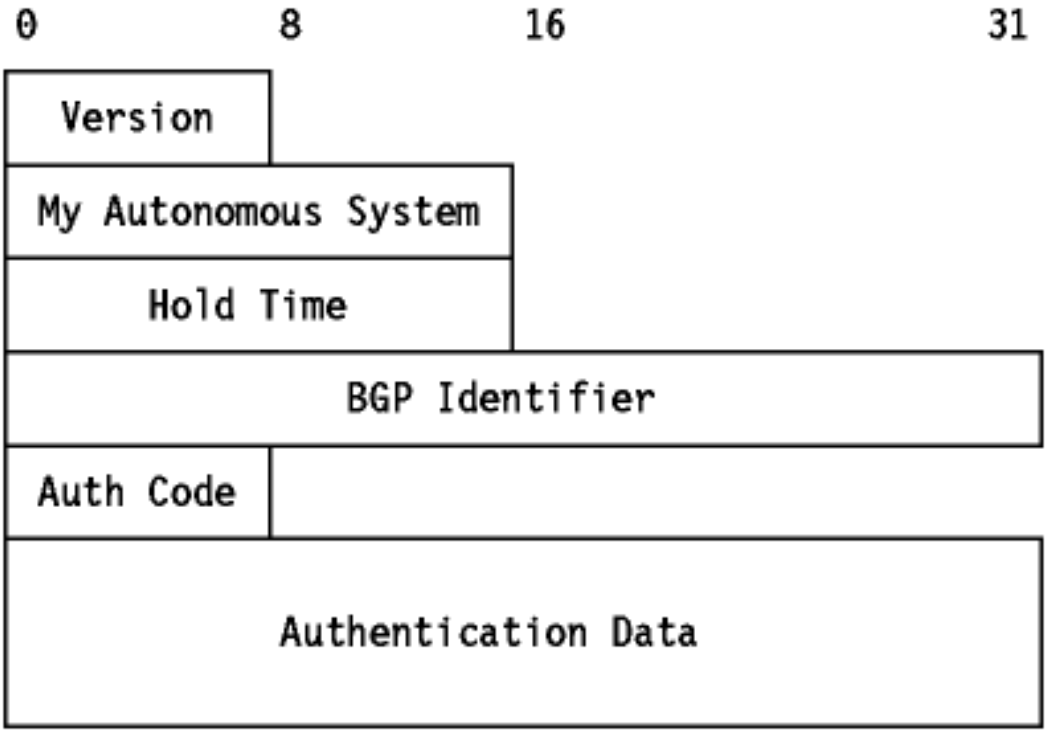


Figura: Mensaje OPEN de BGP-3

Version
: 3 para BGP-3(1 byte)

My Autonomous System
: El número de AS del emisor(2 bytes)

Hold time
: El tiempo máximo en segundos que puede transcurrir entre la recepción de sucesivos mensajes KEEPALIVE y/o UPDATE y/o NOTIFICATION(2 bytes).

BGP Identifier
: Un número de 32 bits único que identifica al BGPS. Es la dirección IP de cualquiera de sus interfaces. Se usa el mismo número para todas las interfaces y vecinos BGP.

Authentication Code
: Define la interpretación de "Authentication Data"(1 byte). BGP-3 sólo define el código de autentificación 0(no hay autentificación).

Authentication Data
: Dependiente de "Authentication Code". La longitud es variable y se deduce a partir de la longitud del mensaje. Para el código 0, el dato se omite.

Los mensajes UPDATE se emplean para transmitir información de encaminamiento. El formato de un mensaje UPDATE se muestra en Figura - Mensaje UPDATE de BGP-3.

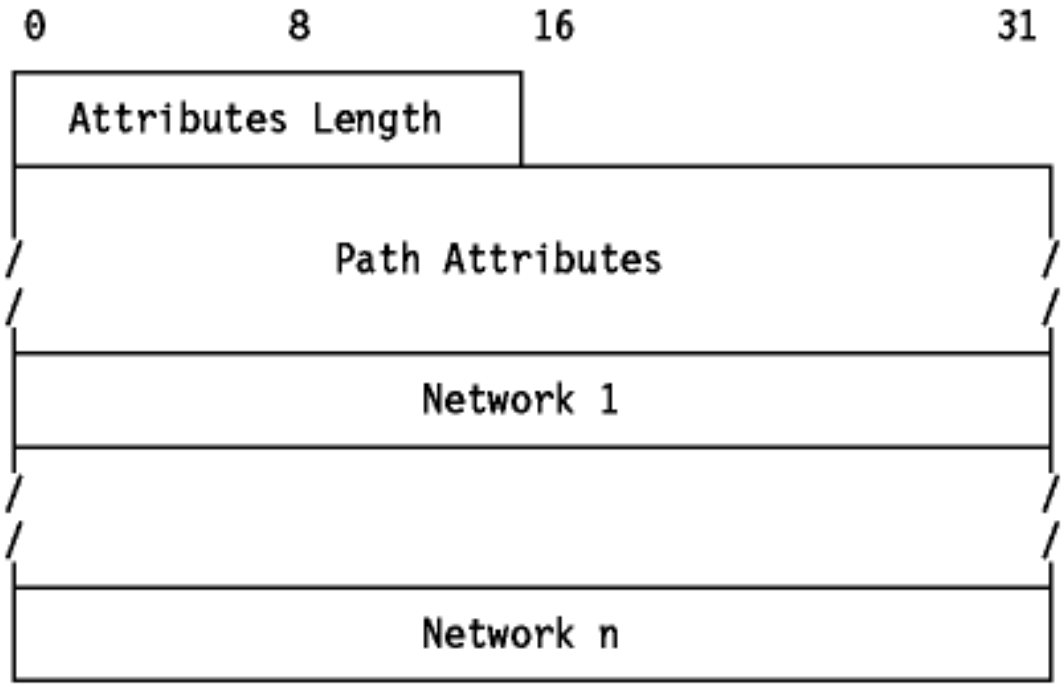


Figura: mensaje UPDATE de BGP-3

Attributes Length
: Longitud del campo "path attributes" bytes (2 bytes).

Path Attributes
: Cada "path attribute"(atributo de la ruta) es una tripleta: < *attribute type, attribute length, attribute value*> donde:

attribute type
: es una campo de 2 bytes, consistente en una byte de flag y un byte de código del tipo de atributo. Los bits del byte de flag son:

X'80'
: Atributo opcional. Si está a uno, el atributo es "*optional"*, si no es bien conocido("*well-known")*. Los atributos bien conocidos son los que todas las implementaciones de BGP-3 deben manejar. Los hay de dos tipos: "*mandatory*", que se deben incluir en cada mensaje UPDATE, y "*discretionary*" que se pueden omitir de los mensajes UPDATE.
Si un BGPs no reconoce un atributo opcional, debería manejarlo según el bit "transitive". Los BGPs pueden actualizar los atributos de los mensajes que retransmiten.

X'40'
: Atributo de tipo "transitive"(transitivo). Debe estar a uno si el atributo es de tipo "mandatory". Para atributos opcionales, si este bit está a uno, el atributo es de tipo "*transitive"*, si no es de tipo "*non-transitive"*. Un atributo "transitive" no reconocido se debe pasar a las consultas de otro BGP después de poner el bit "partial" a uno, y puede ser desechado. Los BGPSs pueden añadir atributos "transitive" de tipo "optional" en un mensaje UPDATE antes de retransmitirlo.

X'20'
: Atributo de tipo "partial". Este bit indica que se pasó un atributo "optional" y "transitive" a un BGPS que no lo reconoció o

que fue añadido por un BGPS distinto del emisor. En todos los demás casos debe ser cero.

X'10'
"Extended length". El campo "attribute length" consta de dos bytes si este bit vale 1, y de uno si es 0. Los cuatro bits de orden inferior son cero y el receptor debe ignorarlos.

Remitirse al RFC 1267 para más detalles.

Los valores *"attribute type code"*(códigos del tipo de atributo) se muestran en Tabla - "Path Attribute" del UPDATE de BGP-3.

Tabla: "Path Attribute" del UPDATE de BGP-3

ORIGIN
El método por el que el AS emisor conoció esta ruta.

0
IGP -- las redes listadas están dentro del AS emisor.

1
EGP -- las redes listadas están fuera del AS emisor y la información de accesibilidad se consiguió por EGP.

2
INCOMPLETE -- las redes listadas se conocieron por otros medios.

AS_PATH
Los números AS de 2 bytes de cada AS en la ruta a la red/es de destino. El número de saltos en la ruta se puede calcular dividiendo al campo "attribute length" por 2.

NEXT_HOP
La dirección IP del ABR que es el siguiente salto en la ruta a la red/es listada/s. Este campo se ignora para las conexiones BGP internas.

UNREACHABLE
Las rutas anunciadas previamente se han convertido en inalcanzables.

INTER-AS METRIC
Este valor se puede usar para elegir entre múltiples rutas a un AS. Si los demás factores son iguales, se elige la ruta con métrica más baja. Este valor se le puede enviar a un BGPS en un AS vecino, y si se recibe sobre una conexión BGP, se puede propagar a través de conexiones BGP internas. Un BGPS no puede retransmitir un INTER-AS METRIC en un mensaje UPDATE al exterior.

attribute length
Longitud(uno o dos bytes).

attribute value
Dependiente del código del valor "attribute type code".

Cada "attribute"(atributo) sólo se puede especificar una vez. El campo "attribute length" determina dónde acaban.

Network 1
El número de red de 32 bits de la primera red descrita en los "path attributes" anteriores. Las subredes y los hosts están inhabilitados explícitamente.

Network n
El número de red de 32 bits de la última red descrita en los "path attributes" anteriores. Las subredes y los hosts están inhabilitados explícitamente. El número de red se puede calcular restando las longitudes de la cabecera BGP-3 y del campo "path attributes" a la longitud del mensaje y dividiendo por 4.

Los mensajes NOTIFICATION se usan para informar al vecino de un error. La conexión BGP se termina tras enviar el mensaje. El formato de este mensaje se muestra Figura - Mensaje NOTIFICATION de BGP-3.

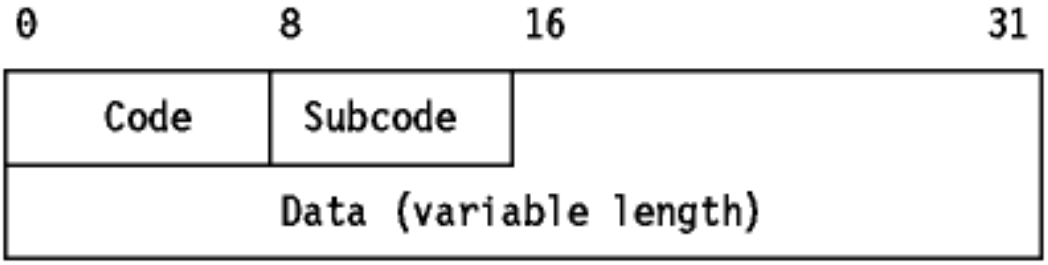


Figura: Mensaje NOTIFICATION de BGP-3

Code

Un byte indicando el tipo de error. Están definidos los siguientes códigos:

1
: Message Header Error

2
: OPEN Message Error

3
: UPDATE Message Error

4
: Hold Timer Expired

5
: Finite State Machine Error

6
: Cease

Subcode
: Un byte de subcódigo que proporciona más información acerca del error. El valor 0 indica que no existen un valor adecuado para este campo. Están definidos los siguientes subcódigos:

Message Header Error Subcodes

1
: Connection Not Synchronized

2
: Bad Message Length

3
: Bad Message Type

OPEN Message Error Subcodes

1
: Unsupported Version Number

2
: Bad Peer AS

3
: Bad BGP Identifier

4
: Unsupported Authentication Code

5
: Authentication Failure

UPDATE Message Error Subcodes

1
: Malformed Attribute List

2
: Unrecognized Well-known Attribute

3
: Missing Well-known Attribute

4
: Attribute Flags Error

5
: Attribute Length Error

6
: Invalid ORIGIN Attribute

7
: AS Routing Loop

8
: Invalid NEXT_HOP Attribute

9
: Optional Attribute Error

10
: Invalid Network Field

Data
: Información de longitud variable dependiente del código y del subcódigo que se pueden emplear para diagnosticar la causa del error. La longitud se puede calcular sustrayendo 21 a la longitud total del mensaje.
Los mensajes KEEPALIVE se emplean para asegurarse de que la conexión sigue activa. Consisten sólo en la cabecera.

Referencias

En los siguientes RFCs se puede encontrar una descripción detallada de BGP-3:

- *RFC 1265 - Análisis del protocolo BGP*
- *RFC 1266 - Experiencia con el protocolo BGP*
- *RFC 1267 - BGP-3*
- *RFC 1268 - Aplicación de BGP en Internet*

3.4.2.2 Interacción de BGP con OSPF

Hay un *protocolo propuesto como estándar* con status electivo que define cómo debería interactuar BGP-3 con OSPF. Cualquier host o "router" que intercambie información dinámicamente entre BGP-3 y OSPF debería adherirse a este estándar. Se describe en el *RFC 1654 - Interacción de BGP con OSPF*.

La interacción de BGP con OSPF cubre la conversión de los campos OSPF en un ELA("External Links Advertisement") al campo "path attributes" de BGP, y vice versa, para tres propiedades de la definición de una ruta.

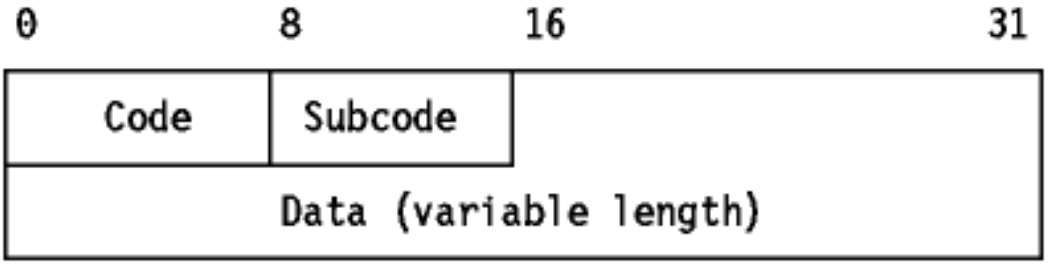


Tabla: Mapeo de campos de BGP a OSPF

El estándar define como se deberían realizar estos mapeos y qué restricciones hay en lo que se podría hacer sistemáticamente.

3.4.2.3 BGP-4 Versión 4 ("Border Gateway Protocol Version 4 ")

BGP-4 es un *protocolo propuesto como est*. Su status es electivo. Se describe en el RFC 1654. Los principales cambios se aplican al soporte de *"supernetting"* o *CIDR("Classless Inter-Domain Routing")* que se describe en CIDR("Classless Inter-Domain Routing"). En particular, BGP-4 soporta prefijos IP y agregación de rutas. Debido a que CIDR es radicalmente distinto de la arquitectura de encaminamiento normal de Internet, BGP-4 es incompatible con BGP-3. Sin embargo, BGP define una mecanismo para que dos BGPSs negocien una versión que ambos entiendan, utilizando el mensaje OPEN. Por lo tanto, es posible implementar BGPSs "bilingües" que permiten la interoperatividad entre BGP-3 y BGP-4.

Los principales cambios de BGP-4 son:

- El número de versión en la cabecera es 4.
- CIDR elimina el concepto de clase de red del encaminamiento inter-dominio, sustituyéndolo por el de prefijo IP.
- La lista de redes en un mensaje UPDATE se sustituye por el *NLRI("Network Layer Reachability Information")*.
- BGP-4 introduce la agregación de múltiples rutas de ASs en entradas únicas o *agregados.* El uso de agregados puede reducir dramáticamente la cantidad de información de encaminamiento requerida.

  Se puede usar un nuevo atributo para una ruta AS(ATOMIC_AGGREGATE) para asegurar que determinados agregados no son desagregados. Otro nuevo atributo(AGGREGATOR) se puede añadirección para agregar rutas con el fin de anunciar qué AS y qué BGPS dentro del AS causaron la agregación .

- BGP-4 modela conceptualmente los datos de un BGPS en tres series de *RIBs("Routing Information Bases")*: uno(Adj-RIBs-In) para los datos obtenidos de vecinos BGP, otro para(Loc-RIB) datos locales obtenidos de las operaciones de las políticas de encaminamiento locales sobre la Adj-RIBs-In, y uno(Adj-RIBs-Out) para datos que han de ser anunciados en mensajes UPDDATE.
- BGP-4 permite la negociación del valor "Hold Time" por cada conexión de modo que los extremos de la misma usen el mismo valor.
- BGP-4 cambia el formato del mensaje UPDATE al formato mostrado enFigura - Mensaje UPDATE de BGP.

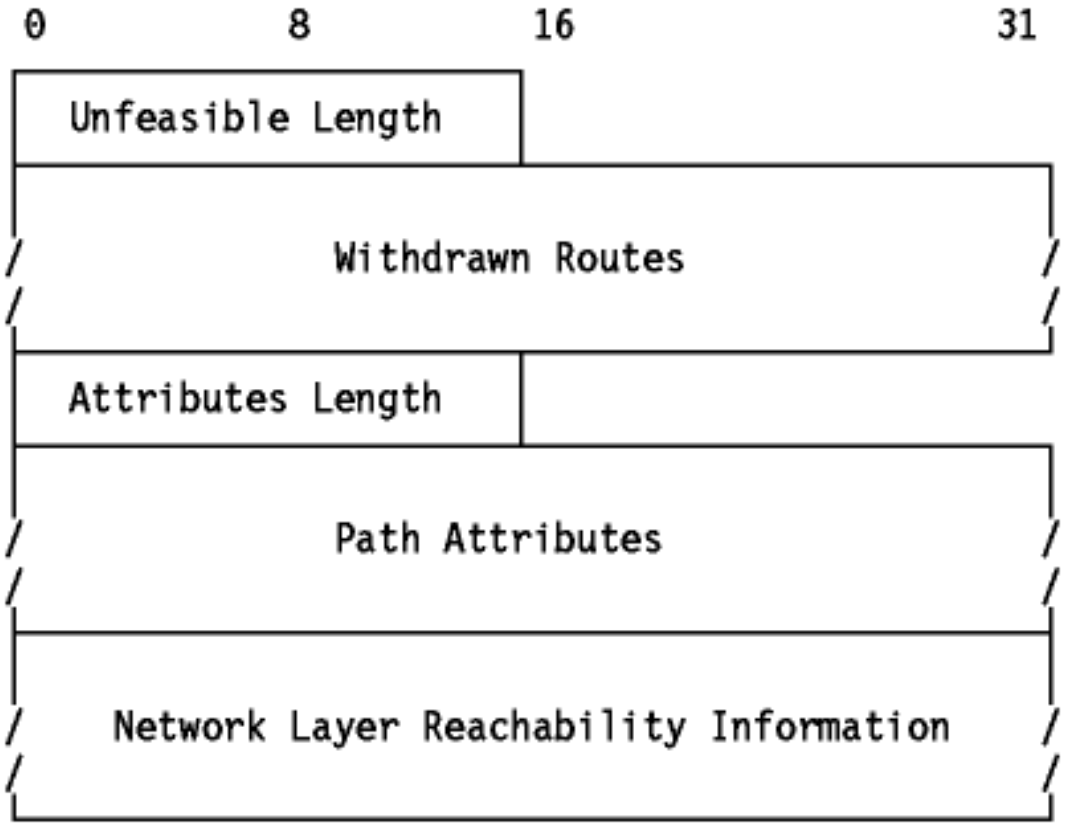


Figura: Mensaje UPDATE de BGP

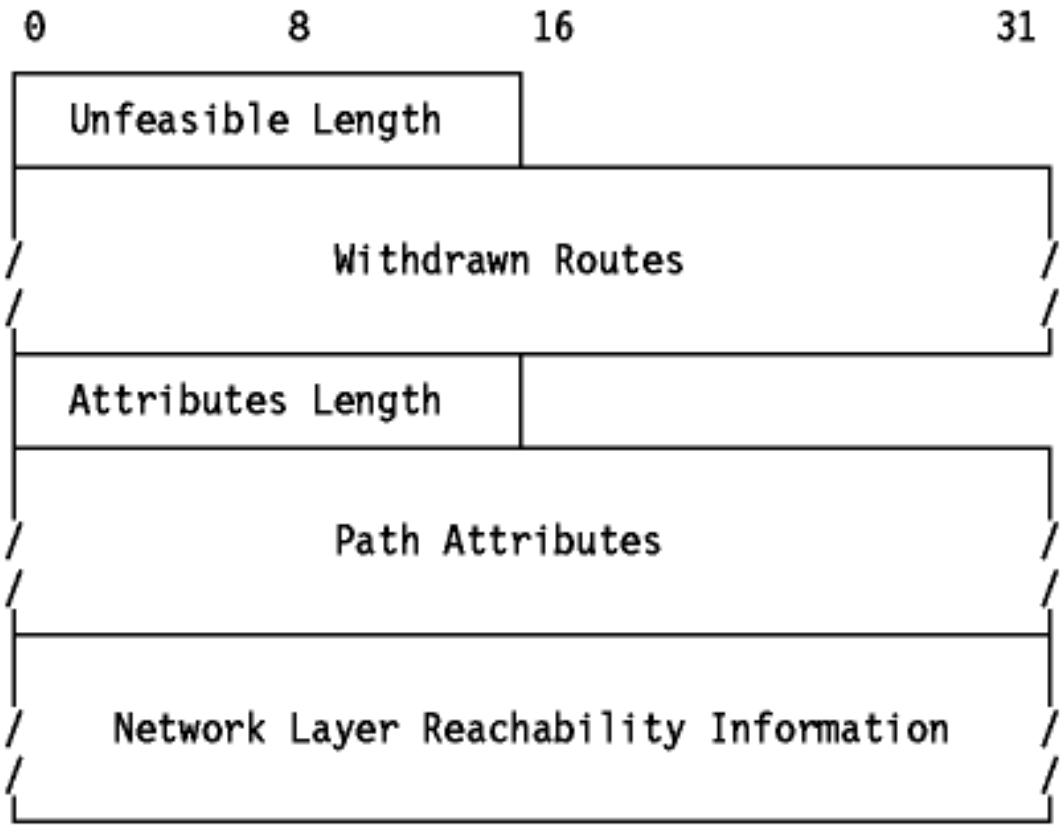
UnfeasibleLength
: Es una contracción de *"Unfeasible Routes Length"* y es un campo de 2 bytes que da la longitud del campo "Withdrawn Routes". Puede ser cero.

Withdrawn Routes
: Una lista de prefijos IP que se están retirando del servicio. Cada entrada tiene la forma *"<length, prefix>"·* donde *length* es un sólo byte que indica la longitud del prefijo en bits, y el prefix es el prefijo IP rellenado hasta el límite con el siguiente byte . Una longitud de cero causa coincidencia con todas las direcciones IP.

Attributes Length
: Un campo de 2 bytes que da la longitud del campo "path attributes" en bytes.

Path Attributes
: Los mismos valores que en BGP-3 excepto en:

attribute type
: Están definidos los siguientes códigos:



ORIGIN
: El método por el que el AS emisor conoció esta ruta.

0
: IGP - las redes listadas están dentro del AS emisor.

1
: EGP - las redes listada están fuera del AS emisor y la información de accesibilidad se obtuvo de EGP.

2
: INCOMPLETE -- las redes listadas se conocieron por otros medios.

AS_PATH
: Una secuencia de ASAPSs(*"AS Path segments")*. Cada uno de ellos tiene la forma *"<path segment type, path segment length, path segment value>"*

path segment type
: Un byte que indica como se ha de interpretar el ASPS.

1
: AS_SET: un conjunto desordenado de aquellos ASs que ha atravesado una ruta para el mensaje UPDATE

2
: AS_SEQUENCE: una secuencia ordenada de los ASs que ha atravesado una ruta para el mensaje UPDATE

path segment length
: Un byte que contiene el número de ASs en el campo "path segment field".

path segment length
: Una serie de números de ASs de 2 bytes.

NEXT_HOP
: La dirección IP del ASBR que es el siguiente salto en la ruta al destino/s listado.

MULTI_EXIT_DISC
: Este valor se puede usar para elegir entre múltiples rutas a un AS. Reemplaza al INTER-AS METRIC de BGP-3. Si todos los demás factores son iguales, se toma la ruta de mínimo valor. Este valor se puede enviar a un BGPS en un AS vecino, y si se recibe sobre una conexión BGP externa, se puede propagar sobre conexiones BGP internas. Un BGPs no puede retransmitir al exterior un valor MULTI_EXIT_DISC que recibe en un mensaje UPDATE.

LOCAL_PREF
: Lo usa un BGPS para informar a otros BGPSs de su propio AS del grado de preferencia de una ruta anunciada.

ATOMIC_AGGREGATE
: Lo usa un BGPS para informar a otros BGPSs que el sistema local seleccionó una ruta menos específica sin seleccionar una más específica que estaba disponible.

AGGREGATOR
: El último número AS que formó el agregado seguido de la dirección IP del BGPS que creó ese agregado.

attribute length
Uno o dos bytes de longitud(según el valor del bit "extended length").
attribute value
Depende del valor de "attribute type code".

Cada atributo sólo se puede especificar una vez. El final de los atributos lo determina el campo "path attributes length".

Network Reachability Information
Una lista de prefijos IP que se están retirando del servicio. Cada entrada tiene la forma *"<length, prefix>"* donde *length* es un sólo byte que indica la longitud del prefijo en bits, y el prefix es el prefijo IP rellenado hasta el límite con el siguiente byte . Una longitud de cero causa coincidencia con todas las direcciones IP.

- Los siguientes subcódigos de error se añaden a la definición del mensaje NOTIFICATION:

OPEN Message Error Subcodes
6
Unacceptable Hold Time
UPDATE Message Error Subcodes
11
Malformed AS_PATH

- BGP-4 define las siguientes reglas para crear AS_PATHs o modificar AS_PATHs conocidos de mano de un vecino BGP.

Las dos siguientes reglas describen como crear un nuevo AS_PATH:

- Cuando se le anuncia a un BGP en el mismo AS, una longitud se crea un AS-PATH de longitud cero.
- Cuando se le anuncia a un BGPS en un AS distinto, el AS _PATH incluye sólo el número AS del emisor. Las siguientes dos reglas describen cómo manejar los AS_PATHs obtenidos de un vecino:
- Al anunciarlo a un BGPS en el mismo AS, el AS-PATH permanece inalterado.
- Cuando se le anuncia a un BGPS del mismo AS, el BGP anunciante añade su propio número AS al comienzo del AS-PATH. Si el AS_PATH comienza con un segmento que es un AS_SEQUENCE, añade su número AS al principio de la secuencia, pero si empieza por AS-SET, el anunciante añade un nuevo segmento del tipo AS_SEQUENCE al comienzo del AS_PATH.

Referencias

En los siguientes RFCs se puede encontrar una detallada descripción de BGP-4:

- *RFC 1654 - BGP-4("Border Gateway Protocol 4")*
- *RFC 1655 - Aplicación de BGP en Internet*
- *RFC 1656 - Mapa de carreteras BGP-4 y experiencias de implementación*

 Tabla de contenidos  Protocolos de aplicación

Capítulo 4. Protocolos de aplicación

Los protocolos de máximo nivel se denominan *protocolos de aplicación*. Se comunican con aplicaciones en otros hosts de internet y constituyen la interfaz de usuario con la pila de protocolos TCP/IP.

4.1 Descripción

4.1.1 Características de las aplicaciones

Todos los protocolos de alto nivel tienen algunas características en común:

- Pueden ser aplicaciones escritas por el usuario o aplicaciones estandarizadas y distribuidas con un producto TCP/IP. De hecho, la pila TCP/IP incluye protocolos de aplicación tales como:
  - TELNET para el acceso interactivo de una terminal a un host remoto.
  - FTP ("File Transfer Protocol") para transferencias de alta velocidad de un disco a otro.
  - SMTP ("simple mail transfer protocol") como sistema de correo de Internet.

  Estas son las aplicaciones implementadas más ampliamente, pero existen muchas otras. Cada imp TCP/IP particular incluye un conjunto más o menos restringido de protocolos de aplicación.

- Usan UDP o TCP como mecanismo de transporte. Recordar que UDP no es fiable ni ofrece control de flujo, por lo que en este caso la aplicación ha de proporcionar sus propia rutinas de recuperación de errores y de control de flujo. Suele ser más fácil desarrollar aplicaciones sobre TCP, un protocolo fiable, orientado a conexión. La mayoría de los protocolos de aplicación utilizan TCP, pero algunas aplicaciones se construyen sobre UDP para proporcionar un mejor rendimiento reduciendo la carga del sistema que genera el protocolo.
- La mayoría de ellas usa el modelo de interacción cliente/servidor.

4.1.2 Modelo cliente/servidor

TCP es un protocolo orientado a conexión. No hay relaciones maestro/esclavo. Las aplicaciones, sin embargo, utilizan un modelo cliente/servidor en las comunicaciones.

Un *servidor* es una aplicación que ofrece un servicio a usuarios de Internet; un *cliente* es el que pide ese servicio. Una aplicación consta de una parte de servidor y una de cliente, que se pueden ejecutar en el mismo o en diferentes sistemas.

Los usuarios invocan la parte cliente de la aplicación, que construye una *solicitud* para ese servicio y se la envía al servidor de la aplicación que usa TCP/IP como transporte.

El servidor es un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones(múltiples clientes) al mismo tiempo.

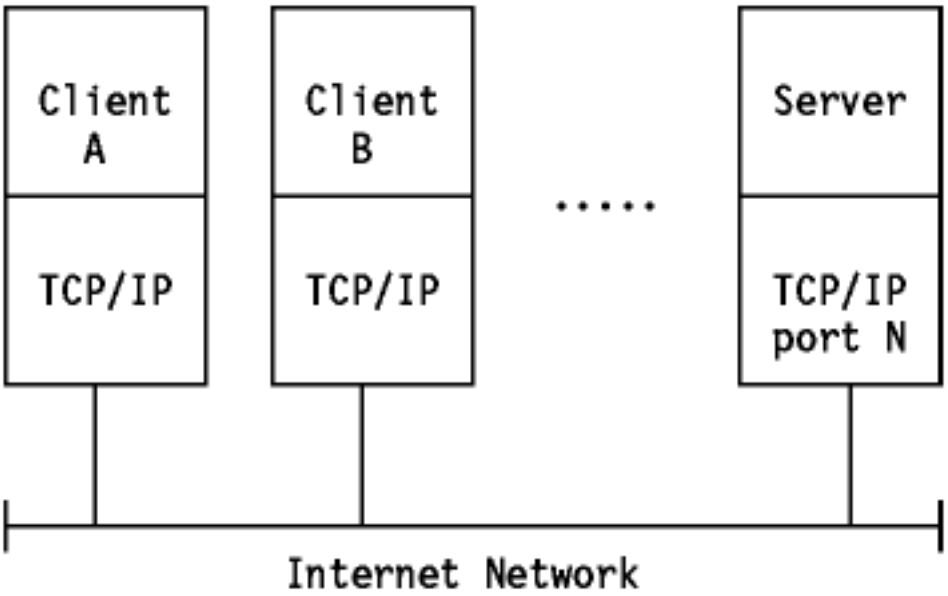


Figura: El modelo de aplicación cliente/servidor

Algunos servidores esperan las solicitudes en *puertos bien conocidos* de modo que sus clientes saben a que zócalo IP deben dirigir sus peticiones. El cliente emplea un puerto arbitrario para comunicarse. Los clientes que se quieren comunicar con un servidor que no usa un puerto bien conocido tienen otro mecanismo para saber a qué puerto dirigirse. Este mecanismo podría usar un servicio de registro como Portmap, que utiliza un puerto bien conocido.

La siguiente sección discute los protocolos de aplicación más extendidos.

Tabla de contenidos TELNET

4.2 TELNET

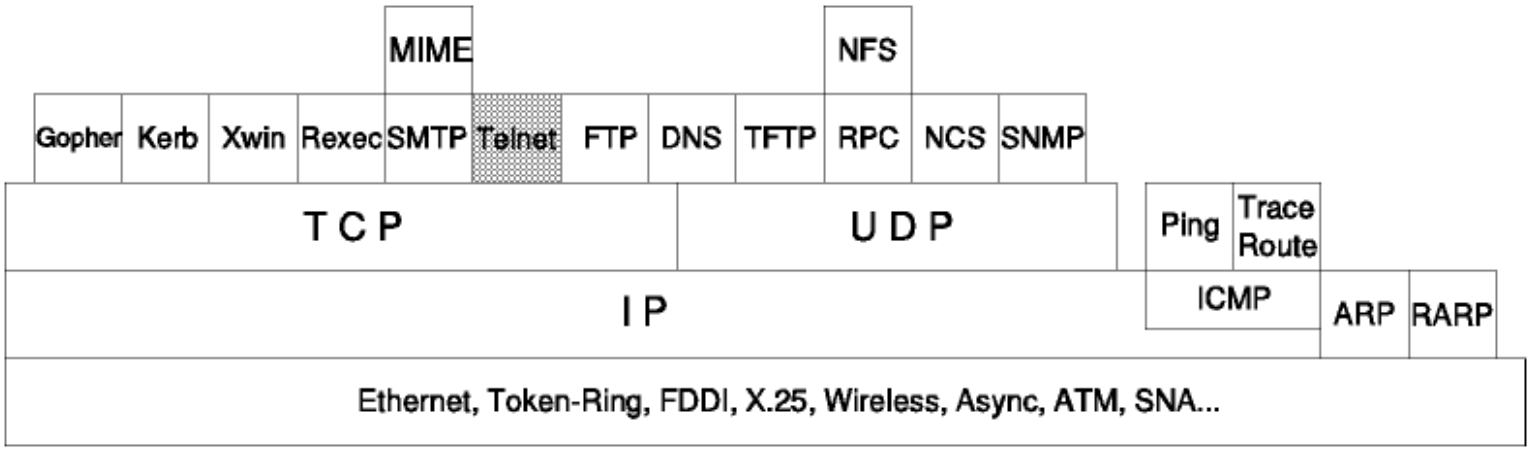


Figura: TELNET- Protocolo de conexión remota.

TELNET es un *protocolo estándar* siendo su número STD de 8. Su status es *recomendado*. Se describe en el *RFC 854 - Especificaciones del protocolo TELNET* y *RFC 855 - "TELNET Option Specifications"*.

El protocolo TELNET proporciona una interfaz estandarizada, a través de la cual un programa de un host(el cliente de TELNET) puede acceder a los recursos de otro host (el servidor de TELNET) como si el cliente fuera una terminal local conectada al servidor.

Por ejemplo, un usuario de una estación de trabajo situada en una LAN se puede conectar al host. Por supuesto, TELNET se puede usar tanto en LANs como en WANs.

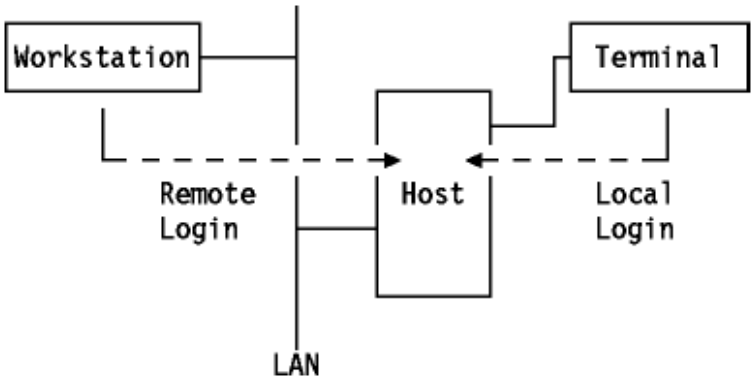


Figura: Conexión remota usando TELNET - TELNET permite la entrada del usuario conectado a la LAN del mismo modo que lo haría el usuario de una terminal local.

La mayoría de las implementaciones de TELNET no soportan entornos gráficos.

4.2.1 Funcionamiento de TELNET

TELNET es un protocolo basado en tres ideas:

- El concepto de *NVT(Network Virtual Terminal) (NVT)*. Una NVT es un dispositivo imaginario que posee una estructura básica común a una amplia gama de terminales reales. Cada host mapea las características de su propia terminal sobre las de su correspondiente NVT, y asume todos los demás hosts harán lo mismo.
- Una perspectiva simétrica de las terminales y los procesos.
- Negociación de las opciones de la terminal. El protocolo TELNET usa el principio de opciones negociadas, ya que muchos host pueden desear suministrar servicios adicionales, más allá de los disponibles en la NVT. Se pueden negociar diversas opciones. El cliente y el servidor utilizan una serie de convenciones para establecer las características operacionales de su conexión TELNET a través de los mecanismos "DO, DON'T, WILL, WON'T"("hazlo, no lo hagas, lo harás, no lo harás") discutidos posteriormente en este capítulo.

Los dos hosts comienzan verificando que existe una comprensión mutua entre ellos. Una vez que se ha completado esta negociación inicial, son capaces de trabajar en el nivel mínimo implementado por la NVT. Después de haber logrado este entendimiento mutuo, pueden negociar opciones adicionales para ampliar las capacidades de la NVT y así reflejar con precisión la capacidad del hardware real que se está usando. Debido al modelo simétrico usado por TELNET, tanto el cliente como el servidor pueden proponer el uso de opciones adicionales.

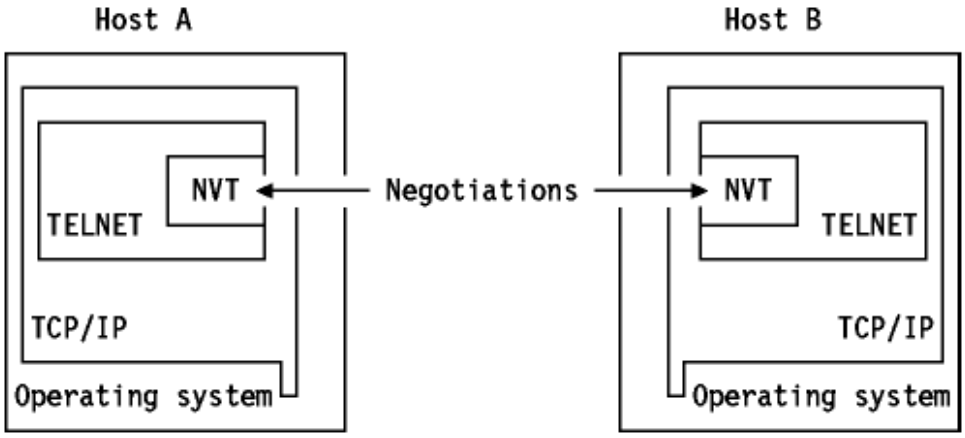


Figura: El modelo simétrico de TELNET - La negociación comienza con la NVT como punto de partida.

4.2.1.1 NVT(Network Virtual Terminal)

La NVT cuenta con un monitor o "display" y un teclado. El teclado produce datos de salida, que se envían por la conexión TELNET. El monitor recibe los datos de entrada que llegan. Las características básicas de una NVT, a menos que sean modificadas por opciones establecidas de común acuerdo, son:

Los datos se representan en código ASCII de 7 bits, transmitido en bytes de 8 bits.

- La NVT es un dispositivo semi-duplex que opera en modo de buffer en línea.
- La NVT proporciona una función de eco local.

Todas estas opciones pueden ser negociadas por los dos hosts. Por ejemplo, se prefiere el eco local porque la carga de la red es inferior y el rendimiento superior pero existe la opción de usar el eco remoto, aunque no se le requiera a ningún host.

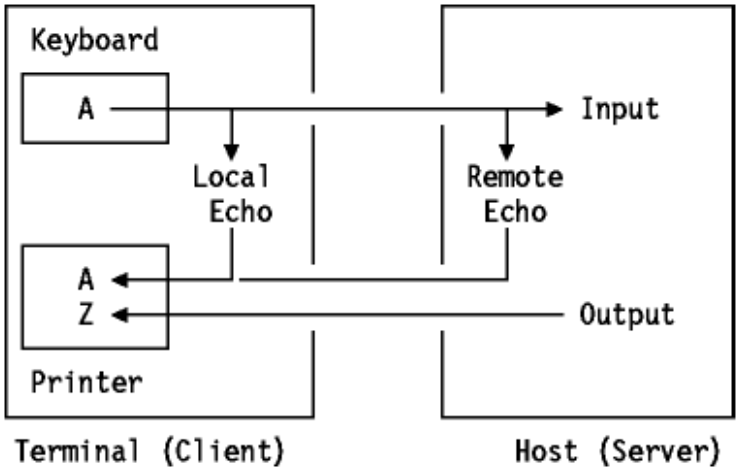
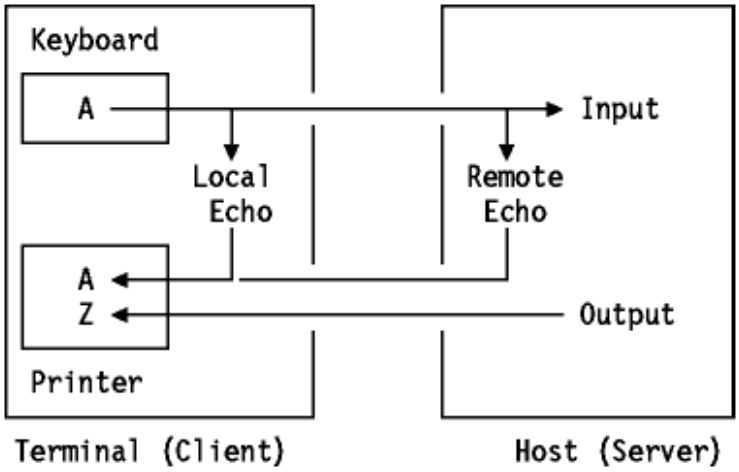


Figura: Opción de eco -Se puede usar la función de eco remoto en vez del local si ambas partes están de acuerdo.

La anchura del retorno de carro y la longitud de la página en un monitor NVT no están especificados. Puede manejar caracteres ASCII imprimibles(códigos ASCII del 32 al 126) y puede entender algunos caracteres ASCII de control tales como:



4.2.1.2 Opiones de TELNET

Hay un gran número de opciones de TELNET; el lector debería consultar *STD 1 - Estándares oficiales de IP* para conocer la estandarización y el status de cada una de ellas. Durante la redacción de este documento, estaban definidas las siguientes opciones:

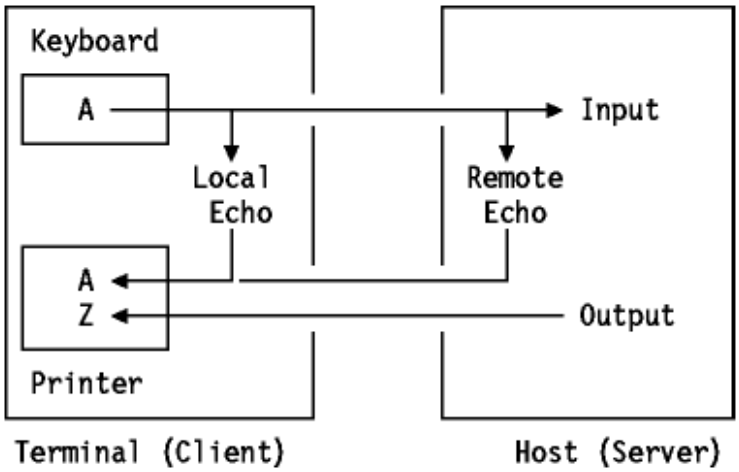


Tabla: Opciones (Parte 1 de 2)

de TELNET

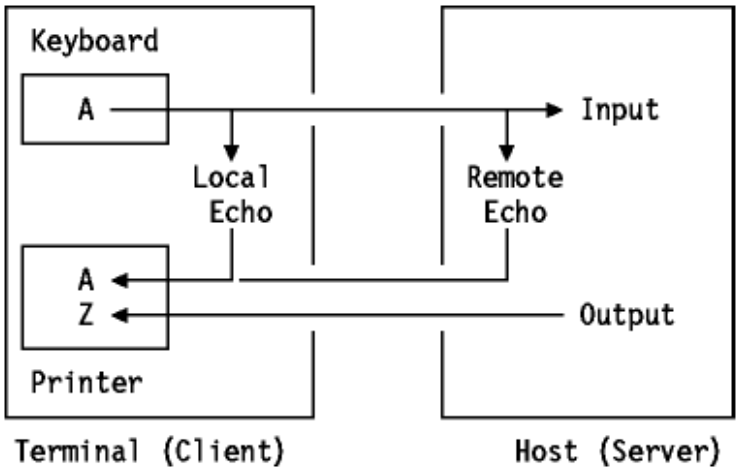


Tabla: Opciones (Parte 2 de 2)

de TELNET Todas las opciones estándar tienen status *recomendado* y el resto, *electivo*. Existe una versión *histórica* de la opción de entorno de TELENET cuyo status es *no recomendado*; es la opción 36 de TELNET y está definida en el RFC 1408.

Posibilidad de uso de la pantalla completa

El TELNET a pantalla completa es posible siempre que el cliente y el servidor tengan medios compatibles para el uso de esta. Por ejemplo, VM y MVS proporcionan un servidor capaz de soportar un TN3270. Para usar este recurso, el cliente debe soportar también el TN3270.

4.2.1.3 Estructura de comandos en TELNET

La comunicación entre cliente y servidor es manejada por comandos internos, que no son accesibles a los usuarios. Todos los comandos internos de TELNET consisten en secuencias de 2 o 3 bytes, dependiendo del tipo de comando.

El carácter IAC("Interpret As Command"; Interpretar Como Comando) es seguido de un código de comando. Si este comando trata con opciones de negociación, el comando tendrá un tercer byte para mostrar el código asociado a la opción indicada.

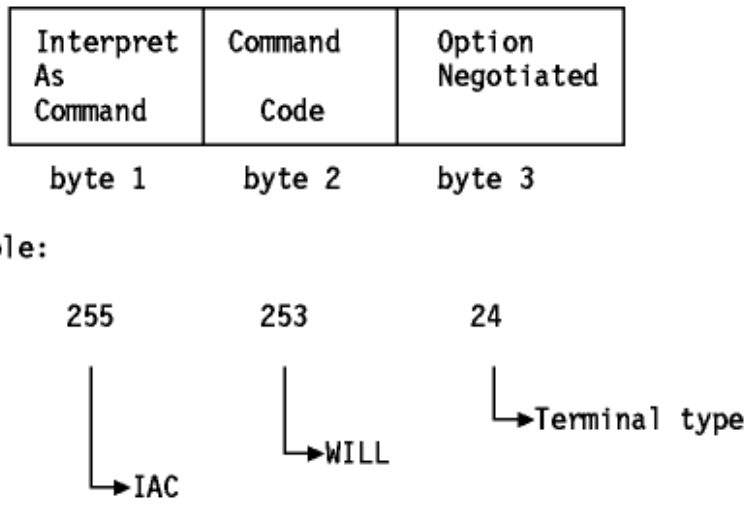
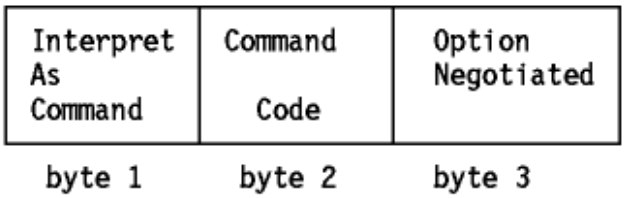
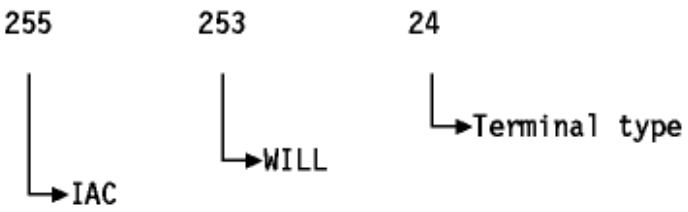


Figura: Estructura de los comandos internos de TELNET - Este comando propone la negociación sobre el tipo de terminal.

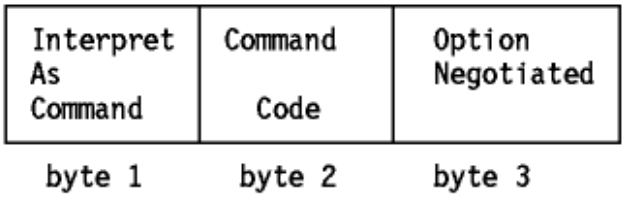


Sample:

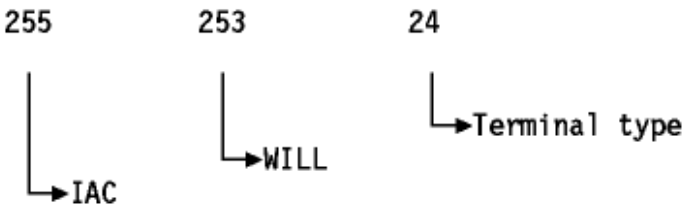


4.2.1.4 Negociación de opciones

Usando los comandos internos, TELNET es capaz de negociar opciones en cada host. La base inicial de la negociación es la NVT: cada host que se quiera conectar debe estar de acuerdo con este mínimo. Cada opción se puede negociar haciendo uso de los cuatro códigos de comando "WILL, WON'T, DO, DON'T" descritos anteriormente. Además, algunas opciones tienen a su vez sub-opciones: si ambas partes acuerdan una opción, usarán los comandos SB y SE para llevar a cabo la sub-negociación. Aquí se muestra un ejemplo simplificado de como funciona la negociación de opciones:



Sample:



Los tipos de terminal se definen en *STD 2 - Números asignados*.

4.2.1.5 Comandos básicos de TELNET

El objetivo principal del protocolo TELNET es proporcionar una interfaz estándar para hosts en una red. Para permitir que comience una conexión. TELNET establece una representación estándar para algunas funciones:

IP
: Interrumpir proceso

AO
: Abortar la salida

AYT
: ¿Estás ahí?

EC
: Borrar carácter

EL
: Borrar línea

SYNCH
: Sincronizar

Tabla de contenidos TFTP("Trivial File Transfer Protocol")

4.3 TFTP(Trivial File Transfer Protocol)

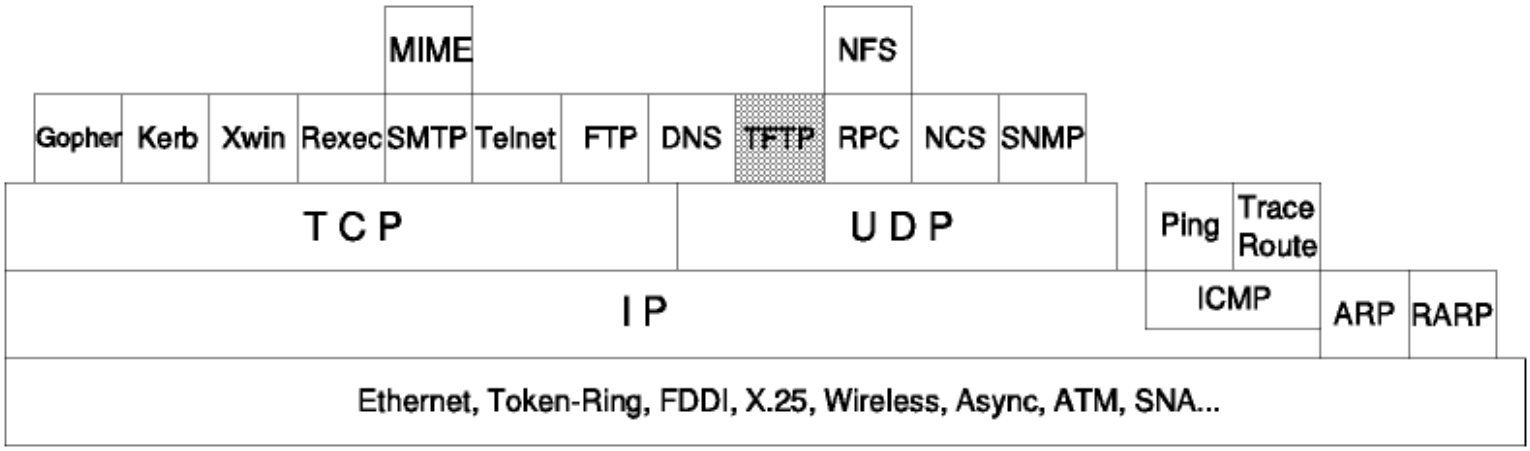


Figura: TFTP(Trivial File Transfer Protocol)

El protocolo TFTP es un *protocolo estándar* siendo su número STD el 33.Su status es electivo y se describe en el RFC 1350 - El protocolo TFTP(Revisión 2).

La transferencia de ficheros en TCP/IP es una transferencia de datos de disco a disco, en oposición, por ejemplo, al comando SENDFILE de VM, una función que en el mundo de TCP/IP se considera de correo, en la que envías a los datos al buzón de alguien(el lector en el caso de VM).

TFTP es un protocolo extremadamente trivial para la transferencia de ficheros. Se implementa sobre la capa UDP(User Datagram Protocol) y carece de la mayoría de las características de FTP (ver FTP(File Transfer Protocol)). La única cosa que es capaz de hacer es leer/escribir un fichero de/en un servidor.

Nota: No dispone de medios para la autentificación de usuarios: es un protocolo inseguro.

4.3.1 Uso

El comando:

```
TFTP <nombre del host>
```

conduce al prompt interactivo en el que se pueden introducir subcomandos:

Connect <host>
    especifica el identificador del host de destino
Mode <ascii/binary>
    especifica el tipo del modo de transferencia
Get <remote filename> [<nombre del fichero local>]
    recupera un fichero
Put <remote filename> [<nombre del fichero local>]
    almacena un fichero
Verbose
    Activa o desactiva el modo "verbose", en el que muestra información adicional durante la transferencia del fichero.
Quit
    salir TFTP

Para una lista completa de estos comandos, consulta tu guía de usuario del TFTP particular que estés usando.

4.3.2 Descripción del protocolo

Cualquier transferencia comienza con una solicitud para leer o escribir un fichero. Si el servidor concede la solicitud, se abre la conexión y el fichero se envía en bloques consecutivos de 512 bytes(longitud fija). Los bloques del fichero se numeran correlativamente, comenzando en 1. Cada paquete de datos debe ser reconocido mediante un paquete de reconocimiento antes de que se envíe el siguiente paquete. Se asume la terminación de la transferencia al recibir un paquete de menos de 512 bytes.

La mayoría de los errores provocarán la terminación de la conexión(falta de fiabilidad), Si un paquete se pierde en la red, se producirá un "timeout", tras el que se efectuará la retransmisión del último paquete (de datos o de reconocimiento).

En el RFC 783 se describió un bug bastante grave, conocido como el Síndrome del Aprendiz de Brujo. Puede causar una retransmisión excesiva en ambas partes de la conexión en algunas circunstancias en las que se producen retardos de red. Se documentó en el RFC 1123 y se corrigió en el 1350. Para más detalles, remitirse a estos RFCs.

4.3.2.1 Paquetes TFTP

Sólo existen cinco tipos de paquetes:

Opcode

    Operación

1
: Read Request(Solicitud de lectura(RRQ))
2
: Write Request(Solicitud de escritura(WRQ))
3
: Data(Datos(DATA))
4
: Acknowledgment(Reconocimiento(ACK))
5
: Error(Error(ERROR))

La cabecera de TFTP contiene el identificador opcode asociado al paquete.

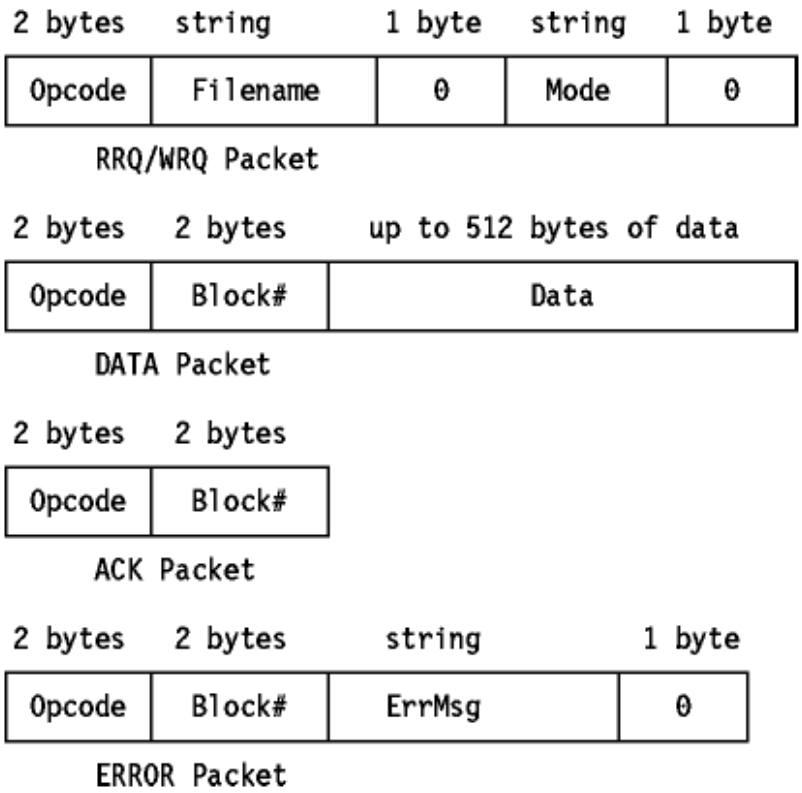


Figura: Paquetes TFTP

4.3.2.2 Modos de transferencia

Actualmente se definen tres modos de transferencia en el RFC 1350:

NetASCII
: US-ASCII tal como se define en el Código Estadounidense Estándar para el Intercambio de Información(*"USA Standard Code for Information Interchange"*) con modificaciones especificadas en el RFC 854 - Especificaciones del protocolo Telnet y extendido para usar el bit de mayor orden. Es decir, se trata de un juego de caracteres de 8 bits, a diferencia del US-ASCII, que es de 7-bits.

Octeto
: También llamado binario, consiste simplemente en bytes de 8 bits.

Correo
: Este modo se definió originalmente en el RFC 783 y el RFC 1350 lo declaró obsoleto. Permitía efectuar la transferencia enviando correo a un usuario en vez de un fichero.

El modo empleado se indica en el paquete "Request for Read/Write(RRQ/WRQ)".

Tabla de contenidos FTP("File Transfer Protocol")

4.4 FTP("File Transfer Protocol")

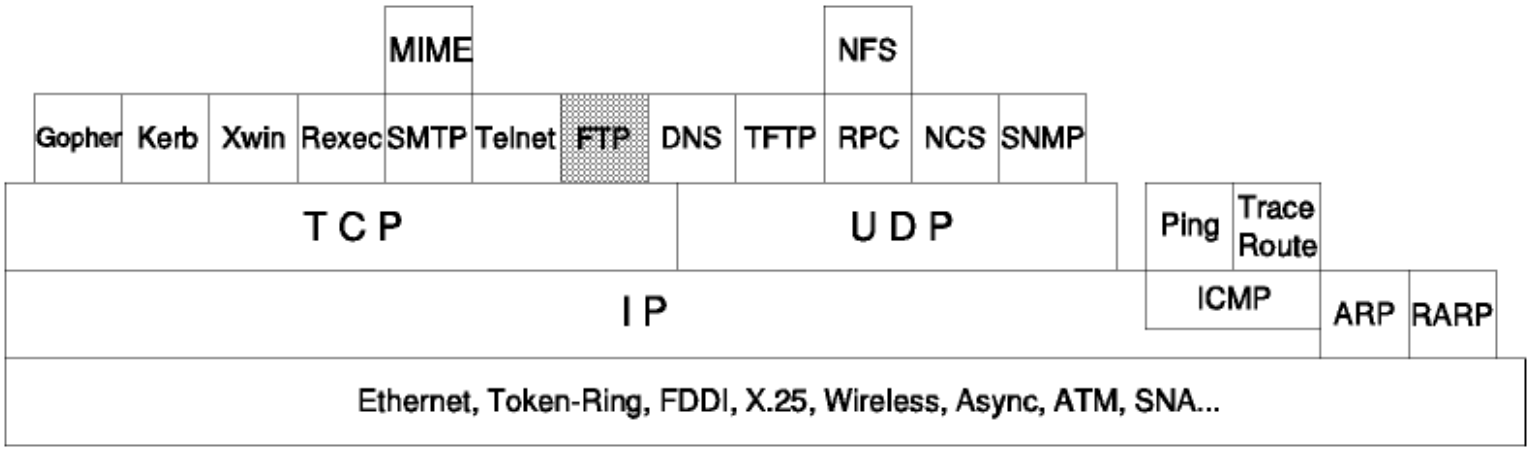


Figura: FTP

FTP es un *protocolo estándar* con el STD 9. Su status es *recomendado*. Se describe en el *RFC 959 - FTP("File Transfer Protocol")*.

La copia de ficheros de una máquina a otra es una de las operaciones más frecuentes. La transferencia de datos entre cliente y servidor puede producirse en cualquier dirección. El cliente puede enviar o pedir un fichero al servidor.

Para acceder a ficheros remotos, el usuario debe identificarse al servidor. En este punto el servidor es responsable de autentificar al cliente antes de permitir la transferencia de ficheros.

Desde el punto de vista de un usuario de FTP, el enlace está orientado a conexión. En otras palabras, es necesario que ambos hosts estén activos y ejecutando TCP/IP para establecer una transferencia de ficheros.

4.4.1 Descripción de FTP

FTP usa TCP como protocolo de transporte para proporcionar conexiones fiables entre los extremos. Se emplean dos conexiones: la primera es para el login y sigue el protocolo TELNET y la segunda es para gestionar la transferencia de datos. Como es necesario hacer un login en el host remoto, el usuario debe tener un nombre de usuario y un password para acceder a ficheros y a directorios. El usuario que inicia la conexión asume la función de cliente, mientras que el host remoto adopta la función de servidor

En ambos extremos del enlace, la aplicación FTP se construye con intérprete de protocolo(PI), un proceso de transferencia de datos, y una interfaz de usuario(ver Figura - Principios de FTP).

La interfaz de usuario se comunica con el PI, que está a cargo del control de la conexión. Este intérprete de protocolo ha de comunicar la información necesaria a su propio sistema de archivos.

En el otro extremo de la conexión, el PI, además de su función de responder al protocolo TELNET, ha de iniciar la conexión de datos. Durante la transferencia de ficheros, los DTPs se ocupan de gestionar la transferencia de datos. Una vez que la operación del usuario se ha completado, el PI ha de cerrar la conexión de control.

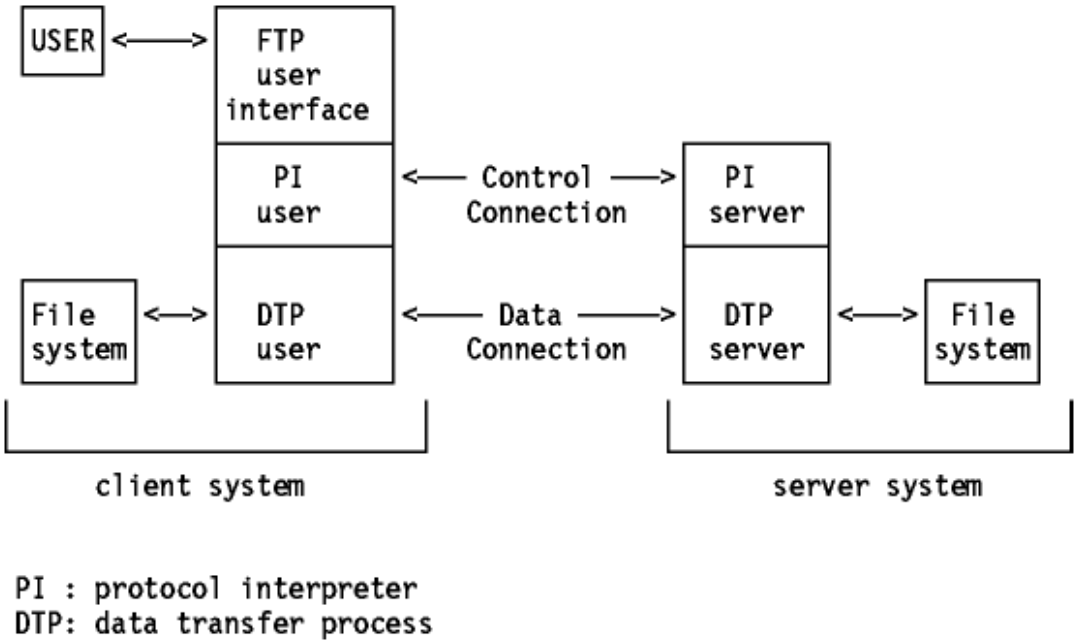


Figura: Principios de FTP

4.4.2 Operaciones de FTP

Al usar FTP, el usuario realizará alguna de las siguientes operaciones:

- Conexión a un host remoto
- Selección de un directorio
- Listado de ficheros disponibles para una transferencia
- Especificación del modo de transferencia
- Copiar ficheros de o a el host remoto
- Desconectar del host remoto

4.4.2.1 Conexión a un host remoto

Para ejecutar una transferencia de ficheros, el usuario comienza haciendo un login en el host remoto. Este es el método primario para manejar la seguridad. El usuario debe tener un identificador y un password para el host remoto, a menos que use un FTP anónimo, descrito en FTP anónimo.

Se usan tres comandos:

Open
: Selecciona el host remoto de inicia la sesión con el login

User
: Identifica al ID del usuario remoto

Pass
: Autentifica al usuario

Site
: Envía información al host remoto utilizado para proporcionar servicios específicos para ese host

4.4.2.2 Selección de un directorio

Cuando se establece el enlace de control, el usuario puede emplear el subcomando cd("change directory") para seleccionar un directorio remoto de trabajo. Obviamente, el usuario sólo podrá acceder a directorios a los que su ID le da acceso. El usuario puede seleccionar un directorio local con el comando lcd("local change directory"). La sintaxis de estos comando depende del sistema operativo.

4.4.2.3 Listado de ficheros disponibles para una transferencia

Se hace con los subcomandos dir o ls.

4.4.2.4 Especificación del modo de transferencia

La transferencia de datos entre sistemas diferentes suele requerir transformaciones de los datos como parte del proceso de transferencia. El usuario ha de decidir dos aspectos de la manipulación de los datos:

- La forma en qué se transferirán los bits.
- Las distintas representaciones de los datos en la arquitectura del sistema.

Esto se controla por medio de dos subcomandos:

Mode
: Especifica si el fichero se ha de tratar como si tuviera estructura de registros o como un flujo de bytes.

Block
: Se respetan las separaciones lógicas entre registros.

Stream
: El fichero se trata como un flujo de bytes. Esta es la opción por defecto, y proporciona una transferencia más eficiente, pero puede que no produzca los resultados deseados cuando se trabaja con un ficheros estructurados por registros.

Type
: Especifica el conjunto de caracteres usado para los datos.

ASCII
: Indica que ambos host están basados en ASCII, o que si uno está basado en ASCII y el otro en EBCDIC, se debería realizar una traducción ASCII-EBCDIC.

EBCDIC
: Indica que ambos host se basan en EBCDIC.

Image
: Indica que los datos deben tratarse como bits contiguos empaquetados en bytes de 8 bits.

Debido a que estos subcomandos no cubren todas las posibles diferencias entre sistemas, el subcomando SITE está disponible para lanzar comandos dependientes del sistema.

4.4.2.5 Copia de ficheros

Get
: Copia un fichero del host remoto al host local.

Put
: Copia un fichero del host local al host remoto.

4.4.2.6 Finalización de la sesión de transferencia

Quit
: Desconecta del host remoto y cierra el FTP. Algunas implementaciones usan el subcomando BYE.

Close
: Desconecta del host remoto pero deja al cliente FTP ejecutándose. Se puede lanzar un comando open para trabajar con otro host remoto.

4.4.3 Códigos de respuesta

Con el fin de gestionar estas operaciones, el cliente y el servidor mantienen un diálogo por medio de TELNET. El cliente lanza comandos, y el servidor contesta con *códigos de respuesta*. Las respuestas incluyen también comentarios para el usuario, pero el cliente usa sólo los códigos.

Los códigos de respuesta tienen tres dígitos, siendo el primero el más significante.

Reply code	Description
1xx	Positive preliminary reply.
2xx	Positive completion reply.
3xx	Positive intermediate reply.
4xx	Transient negative completion reply.
5xx	Permanent negative completion reply.

Tabla: Códigos de respuesta de FTP - Los dígitos primero y segundo proporcionan más detalles de la respuesta.

Ejemplo

Para comando de usuario, mostrado **así**, el servidor FTP responde con un mensaje que comienza con un código de 3 dígitos, mostrado así:

```
FTP foreignhost
220 service ready
USERNAME cms01
331 user name okay
PASSWORD xyxyx
230 user logged in
TYPE Image
200 command okay
```

4.4.5 Ejemplo de una sesión FTP

```
[C:\SAMPLES]ftp host01.itsc.raleigh.ibm.com
Connected to host01.itsc.raleigh.ibm.com.
220 host01 FTP server (Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
Name (rs60002): cms01
331 Password required for cms01.
Password: xxxxxx
230 User cms01 logged in.
ftp> put file01.tst file01.tst
200 PORT command successful.
150 Opening data connection for file01.tst (1252 bytes).
226 Transfer complete.
local: file01.tst remote: file01.tst
1285 bytes received in 0.062 seconds (20 Kbytes/s)
ftp> close
221 Goodbye.
ftp> quit
```

Figura: ejemplo de una sesión FTP - Transferencia de un fichero a un host remoto

4.4.5.1 FTP anónimo

Muchos sitios TCP/IP implementan lo que se conoce como *FTP anónimo*, lo que significa que permiten el acceso público a los ficheros de algunos directorios. El usuario remoto sólo tiene que usar el ID *anonymous* y el password *guest* o alguna otra convención de password, por ejemplo el identificador de usuario para el E-mail. La convención que usa cada sistema se le explica al usuario durante el proceso de login.

Tabla de contenidos DNS ("Domain Name System")

4.5 DNS("Domain Name System")

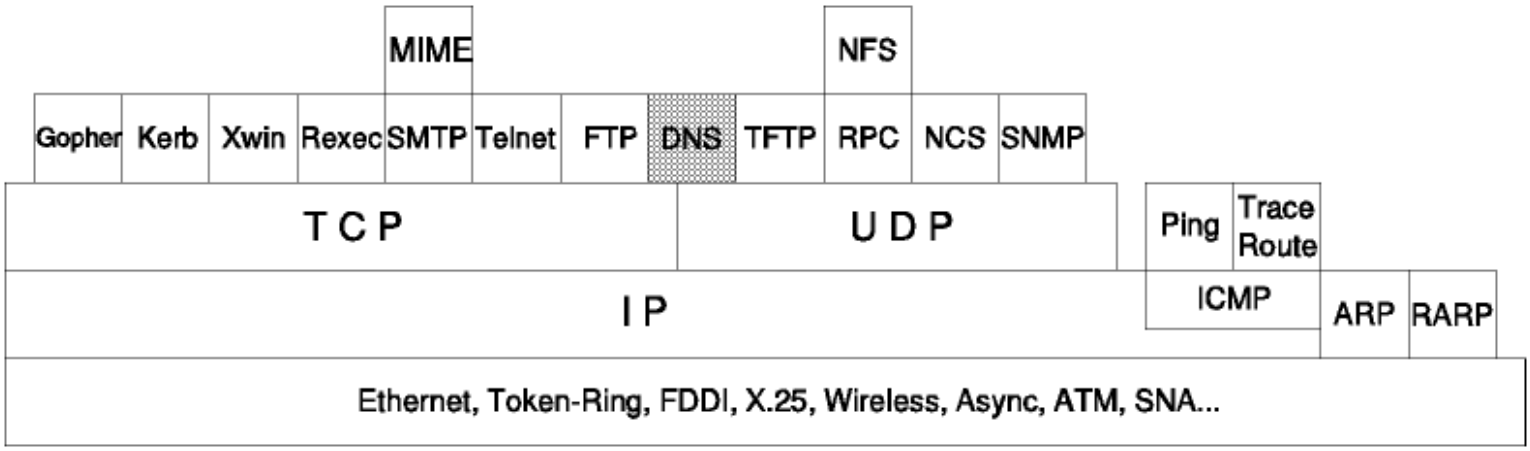


Figura: El DNS

El DNS es un *protocolo estándar* con STD 13. Su status es *recomendado*. Se describe en los RFCs 1034 y 1035. Esta sección explica la implementación del DNS y de los servidores de nombres. Ver "Domain Name System" para una descripción del DNS y su relación con el esquema de direccionamiento IP.

4.5.1 El espacio de nombres distribuido

El DNS usa el concepto de *espacio de nombres distribuido*. Los nombres simbólicos se agrupan en *zonas de autoridad*, o más comúnmente, *zonas*. En cada una de estas zonas, uno o más hosts tienen la tarea de mantener una base de datos de nombres simbólicos y direcciones IP y de suministrar la función de servidor para los clientes que deseen traducir nombres simbólicos a direcciones IP. Estos *servidores de nombres* locales se interconectan lógicamente en una árbol jerárquico de *dominios*. Cada zona contiene una parte del árbol o *subárbol* y los nombres de esa zona se administran con independencia de los de otras zonas. La autoridad sobre zonas se delega en los servidores de nombres. Normalmente, los servidores de nombres que tienen autoridad en zona tendrán nombres de dominio de la misma, aunque no es imprescindible. En los puntos en los que un dominio contiene un subárbol que cae en una zona diferente, se dice que el servidor / servidores de nombres con autoridad sobre el dominio superior *delegan autoridad* al servidor / servidores de nombres con autoridad sobre los subdominios. Los servidores de nombres también pueden delegar autoridad en sí mismos; en este caso, el espacio de nombres sigue dividido en zonas, pero la autoridad para ambas las ejerce el mismo servidor . La división por zonas se realiza utilizando registros de recursos guardados en el DNS:

Registro SOA("Start of Authority")
Define el inicio de una zona
Registro NS("Name Server")
Marca el fin de una zona iniciada por un SOA y apunta a un servidor de nombres con autoridad sobre la zona siguiente

En este contexto, el comienzo de un dominio está más cerca a la raíz del árbol que a sus terminaciones. En la raíz, no puede haber servidores de nombres superiores para delegar autoridad: la autoridad para la raíz se deposita en un conjunto de *servidores de nombres de la raíz*. (11)

El resultado de este esquema es:

- En vez de tener un servidor central para la base de datos, el trabajo implicado en mantenerla se reparte entre los hosts a lo largo y ancho del espacio de nombres.
- La autoridad para crear y cambiar nombres simbólicos de hosts y la responsabilidad de mantener una base de datos para ellos le corresponde a la organización propietaria de la zona que los contiene.
- Desde el punto de vista del usuario, hay una sola base de datos que trata la resolución de las direcciones.

Nota: Aunque los dominios dentro del espacio de nombres se mapean con frecuencia a redes y subredes con el mecanismo de direccionamiento IP, este no es un requisito del DNS. Considerar un "router" entre dos subredes: tiene dos direcciones IP, una por cada adaptador, pero normalmente no tiene por qué poseer dos nombres simbólicos.

4.5.2 Resolución de dominios

La resolución de nombres de dominio es un proceso cliente/ servidor . La función del cliente(el *"resolver"* o *"name resolver"*) es transparente al usuario y la llama una aplicación para mapear nombres de alto nivel a direcciones IP o viceversa. El servidor de nombres(llamado también *servidor de nombres de dominio*) es una aplicación servidora que traduce los nombres de alto nivel de las máquinas a direcciones IP. El proceso básico se muestra en Figura - Usando un "full resolver" para la resolución de nombres de dominio y Figura - Usando una "stub resolver" para la resolución de nombres de dominio. El primero muestra un programa denominado *"full resolver"*, distinto del programa de usuario , que envía todas las peticiones al servidor de nombres . El servidor de nombres cachea las respuestas para su uso en el futuro; posiblemente será él mismo el que las use. El último muestra un *"stub resolver"*, que es una rutina enlazada con el programa de usuario, que envía las peticiones a un servidor de nombres. El servidor de nombres suele cachear las respuestas, aunque esto depende de la implementación. En UNIX, el "stub resolver" se implementa con dos funciones de librería: gethostbyname() y gethostbyaddr() para convertir nombres de hosts a direcciones IP y viceversa. Otras plataformas tienen rutinas iguales o parecidas. Los "stub resolvers" son mucho más comunes que los "full resolvers".

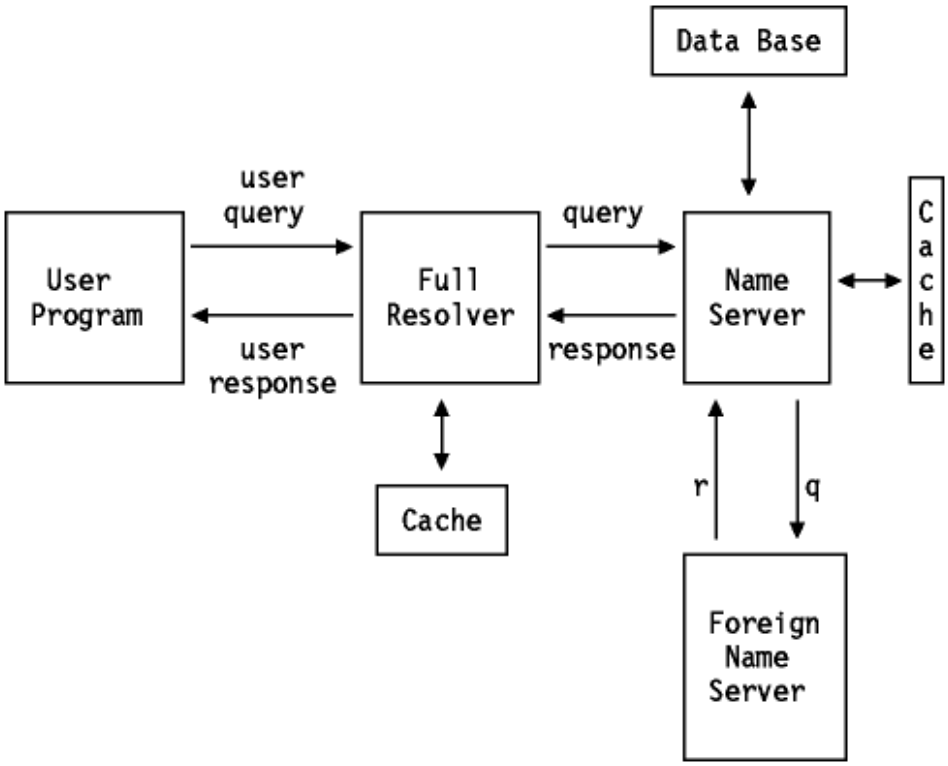


Figura: Usando una "full resolver" para la resolución de nombres de dominio

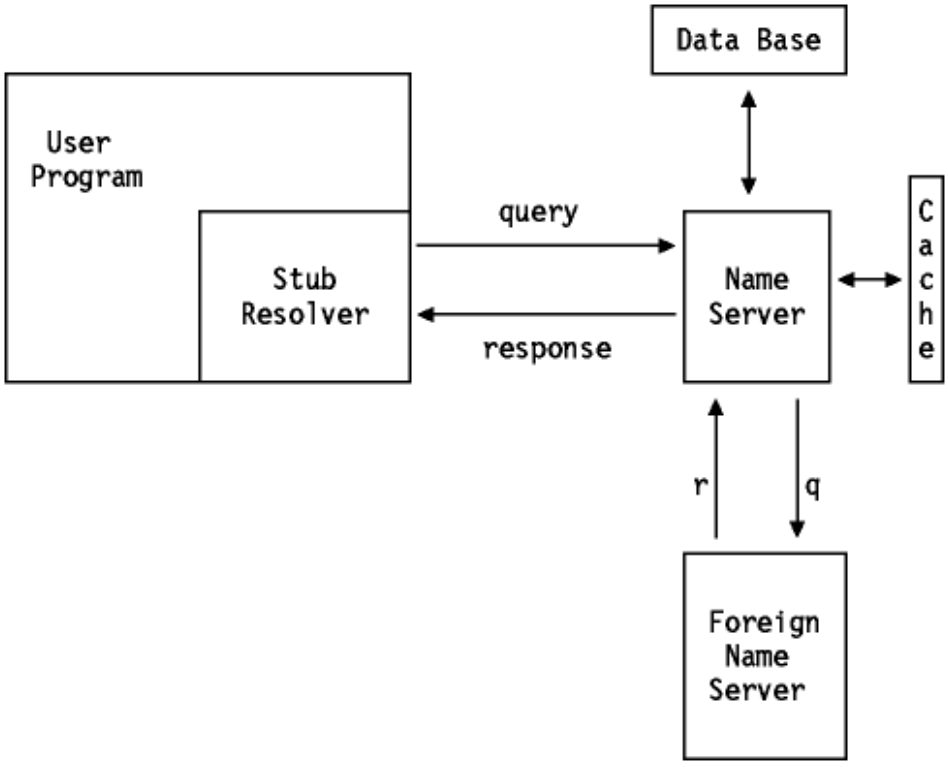


Figure: Usando una "stub resolver" para la resolución de nombres de dominio

4.5.2.1 Funcionamiento del "resolver" de nombres de dominio

Las peticiones sobre nombres de dominio pueden ser de dos tipos: *recursivas* o *iterativas* (llamadas también *no-recursivas*). Un bit de flag en la consulta especifica si el cliente desea una consulta recursiva y un bit de flag en la respuesta indica si el servidor soporta peticiones recursivas. La diferencia entre una consulta recursiva y una iterativa aparece cuando el servidor recibe una solicitud a la que por sí mismo no puede dar una respuesta completa. Una consulta recursiva demanda que el servidor lance a su vez una consulta para determinar la información buscada y luego devolvérsela al cliente. Una consulta iterativa implica que el servidor de nombres debería devolver la información de la que disponga además de una lista de servidores adicionales con los que el cliente puede contactar para completar su consulta.

Las respuestas de nombres de dominio pueden ser de dos tipos: *autoritativas* y *no-autoritativas*. Un bit de flag en la respuesta indica de qué tipo es la respuesta. Cuando un servidor de nombres recibe una consulta para un dominio en una zona en la que tiene autoridad, devuelve una respuesta con el bit de flag activo. Si no tiene autoridad en esa zona, su reacción depende de si el flag de recursividad está o no activo.

- Si el flag de recursividad está activo y el servidor la soporta, dirigirá su consulta a otro servidor de nombres. Este será un servidor con autoridad sobre el dominio de la consulta, o uno de los servidores de nombres de la raíz. Si el segundo servidor no devuelve una respuesta autoritativa, el proceso se repite.

  Cuando un servidor(o un "full resolver") recibe una respuesta, lo cacheará para mejorar el rendimiento de consultas repetidas. La entrada de la cache se almacena con un tiempo máximo especificado en la respuesta en un campo *TTL("time to live")* de 32 bits. 172,800 segundos(dos días) es un valor típico.

- Si el flag de recursividad no está activo o el servidor no soporta consultas recursivas, devolverá la información que tenga en su caché y una lista de servidores capaces de dar respuestas autoritativas.

4.5.2.2 Funcionamiento del servidor de nombres de dominio

Cada servidor de nombres tiene *autoridad* para cero o más zonas. Hay tres tipos de servidor de nombres:

primario
: Un servidor de nombres primario carga de disco la información de una zona, y tiene autoridad sobre ella.

secundario
: Un servidor de nombres secundario tiene autoridad sobre una zona, pero obtiene la información de esa zona de un servidor primario utilizando un proceso llamado *transferencia de zona*. Para permanecer sincronizado, los servidores de nombres secundarios consultan a los primarios regularmente(típicamente cada tres horas) y reejecutan la transferencia de zona si el primario ha sido actualizado. Un servidor de nombres puede operar como primario o secundario para múltiples dominios, o como primario para unos y secundario para otros. Un servidor primario o secundario realiza todas las funciones de un servidor caché.

caché
: Un servidor de nombres que no tiene autoridad para ninguna zona se denomina servidor caché. Obtiene todos sus datos de servidores primarios o secundarios. Requiere al menos un registro NS para apuntar a un servidor del que pueda obtener la información inicialmente.

Cuando un dominio se registra en la raíz y se establece una zona de autoridad separada, se aplican las siguientes reglas:

- El dominio se debe registrar en el administrador de la raíz
- Debe haber un administrador identificado para el dominio
- Debe haber al menos dos servidores de nombres con autoridad para la zona que sean accesibles desde fuera y dentro del dominio para evitar cualquier posible punto débil

También se recomienda que los servidores de nombres que delegan autoridad apliquen estas reglas, ya que son responsables del comportamiento de los servidores de nombres delegados.

4.5.3 Registros de recursos del DNS

La base de datos distribuida del DNS se compone de *RRs("resource records")*. Estos proporcionan una mapeado entre nombres de dominio y *objetos de red*. Los objetos de red más comunes son las dirección de los hosts, pero el DNS está diseñado para acomodarse a una variada gama de distintos objetos. El formato general del registro de recurso es:

```
name ttl class type rdata
```

donde:

name
: Es el nombre de dominio a definir. El DNS es muy general en las reglas de composición de nombres de dominio. Sin embargo, se recomienda una sintaxis para crearlos que minimiza la probabilidad de que las aplicaciones que usen un "resolver"(es decir, la mayoría de las aplicaciones TCP/IP) los malinterpreten. Un nombre de dominio que siga esta sintaxis consistirá en una serie de etiquetas formadas por caracteres alfanuméricos o guiones, cada etiqueta con una longitud de 1 a 63 caracteres, comenzando con un carácter alfabético. Cada par de etiquetas está separado por un punto en forma legible para el ojo humano, pero no en la misma forma que se usa dentro de los mensajes DNS. Los nombres de dominio no son sensibles a mayúsculas y minúsculas.

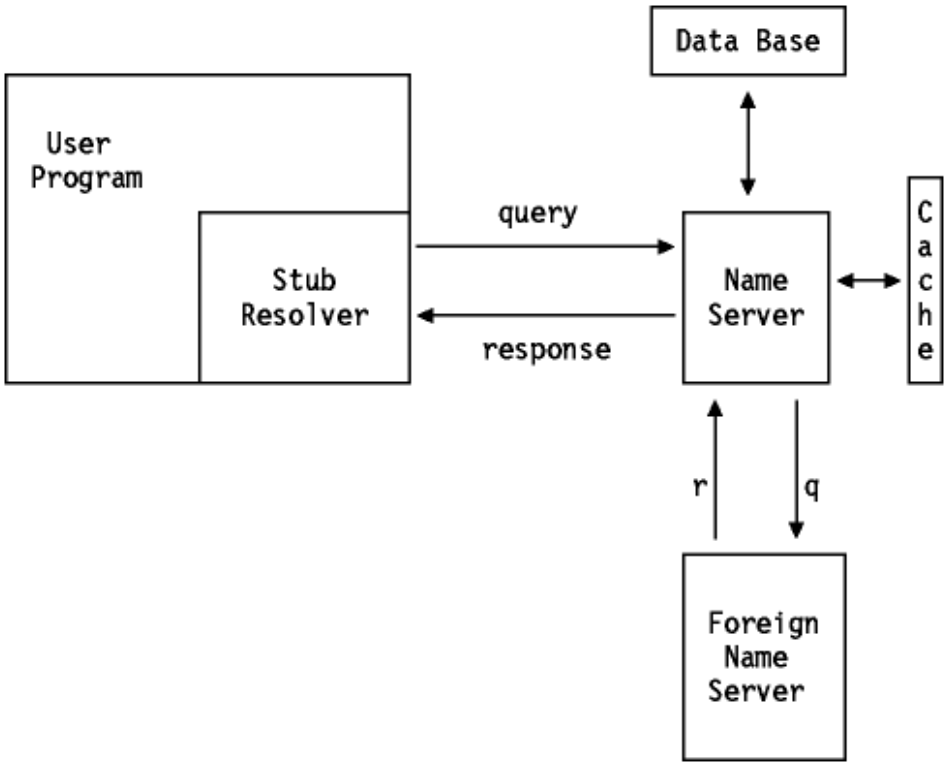
ttl
: Es el *"time-to-live"* o tiempo en segundos que el registro será válido en la caché de un servidor de nombres. Se almacena en el DNS como un valor de 32 bits sin signo. 86400(un día) es un valor típico para registros que apuntan a una dirección IP.

class
: Identifica la familia del protocolo. Valores habituales son:

IN
: Sistema de Internet

CH
: Sistema Chaos

type
: Identifica el tipo de recurso del registro.
Los diferentes tipos se describen en detalle en los RFCs 1034, 1035 y 1706. Cada tipo tienen un nombre y un valor. Valores corrientes incluyen:



Rdata
: El valor depende del tipo, por ejemplo:

A
: Un dirección IP de 32 bits(si la clase es IN)

CNAME
: Un nombre de dominio

MX
: Un valor por defecto de 16 bits(se prefieren valores bajos) seguido de un nombre de dominio.

NS
: Un nombre de host.

PTR
: Un nombre de dominio.

4.5.4 Mensajes del DNS

Todos los mensajes del DNS utilizan un único formato. Este formato se muestra en: Figura - Formato del mensaje DNS. El "resolver" envía la trama al servidor de nombres. Sólo la cabecera y la sección "question" se utilizan para la consulta. Las respuestas o retransmisiones de las consultas usan la misma trama, pero llenan más secciones de la misma(las secciones "answer/authority/additional").

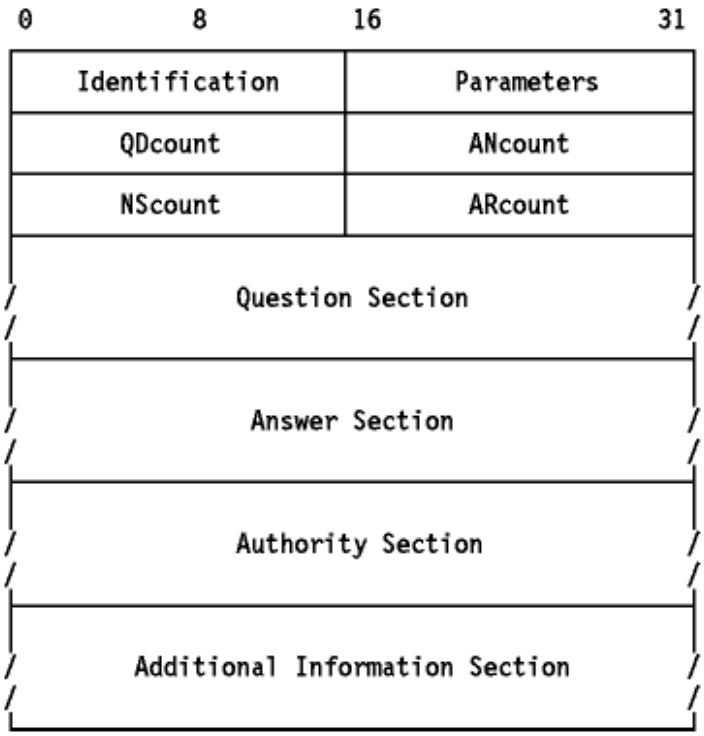


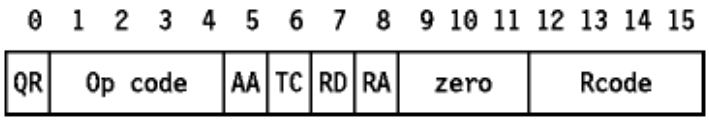
Figure: Formato del mensaje DNS

4.5.4.1 Formato de la cabecera

La sección de cabecera siempre ha de aparecer y tiene una longitud fija de 12 bytes. Las otras secciones son de longitud variable.

ID
: Un identificador de 16 bits asignado por el programa. Este identificador se copia en la respuesta correspondiente del servidor de nombres y se puede usar para diferenciar respuestas cuando concurren múltiples consultas.

Parameters
: Un valor de 16 bits con el siguiente formato:



QR
: Flag que indica consulta(0) o respuesta(1)

Op code
: Campo de 4-bit especificando el tipo de consulta:

0
: consulta estándar(QUERY)

1
: consulta inversa(IQUERY)

2
: solicitud del estado del servidor(STATUS) Se reservan los otros valores para su uso en el futuro

AA
: Flag de respuesta autoritativa. Si está activo en una respuesta, especifica que el servidor de nombres que responde tiene autoridad para el nombre de dominio enviado en la consulta.

TC
: Flag de truncado. Activo si el mensaje es más largo de lo que permite el canal.

RD
: Flag de recursividad. Este bit indica al servidor de nombres que se pide resolución recursiva. El bit se copia en la respuesta.

RA
: Flag de recursividad disponible. Indica si el servidor de nombres soporta resolución recursiva.

zero
: 3 bits reservados para uso futuro. Deben ser cero.

Rcode
: Código de respuesta de 4 bits. Posibles valores son:

0
: Ningún error.

1
: Error de formato. El servidor fue incapaz de interpretar el mensaje.

2
: Fallo en el servidor. El mensaje no fue procesado debido a un problema con el servidor.

3
: Error e nombre. El nombre de dominio de la consulta no existe. Sólo válido si el bit AA está activo en la respuesta.

4
: No implementado. El tipo solicitado de consulta no está implementado en el servidor de nombres.

5
: Rechazado. El servidor rechaza responder por razones políticas. Los demás valores se reservan para su usuario en el futuro.

QDcount
: Un entero sin signo de 16 bits que especifica el número de entradas en la sección "question".

ANcount
: Un entero sin signo de 16 bits que especifica el número de RRs en la sección "answer".

NScount
: Un entero sin signo de 16 bits que especifica el número de RRs en la sección "authority".

ARcount
: Un entero sin signo de 16 bits que especifica el número de RRs en la sección "additional records".

4.5.4.2 Sección "Question"

La siguiente sección contiene las consultas al servidor de nombres. Contiene QDcount(generalmente 1) entradas, cada una con el formato mostrado en Figure - Formato del campo "Question".

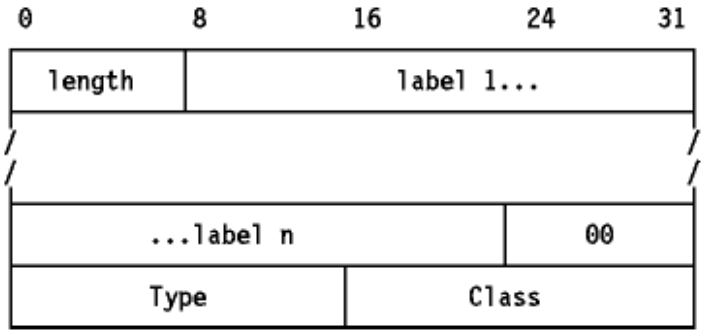


Figura: Formato del campo "Question" - Todos estos campos están alineados por bytes. La alineación del campo Type a 4 bytes es un ejemplo, y no es obligatoria en el formato.

length
: Un byte que indica la longitud de la siguiente etiqueta.

label
: Un elemento del nombre de dominio. El nombre de dominio se almacena como una serie de etiquetas de longitud variable, cada una precedida por un campo "length".

00
: X'00' indica el fin del dominio y representa la etiqueta nula del dominio raíz.

Type
: 2 bytes especificando el tipo de consulta. Puede tener cualquier valor del campo "Type" del registro.

Class

2 bytes especificando la clase de consulta. Para consultas en Internet, será "IN".

4.5.4.3 Secciones "Answer", "Authority" y "Additional Resource"

Estas tres secciones contienen un número variable de registros de recursos. El número se especifica en el campo correspondiente de la cabecera. Los registros tienen el formato mostrado en Figura - Formato de la sección "Answer".

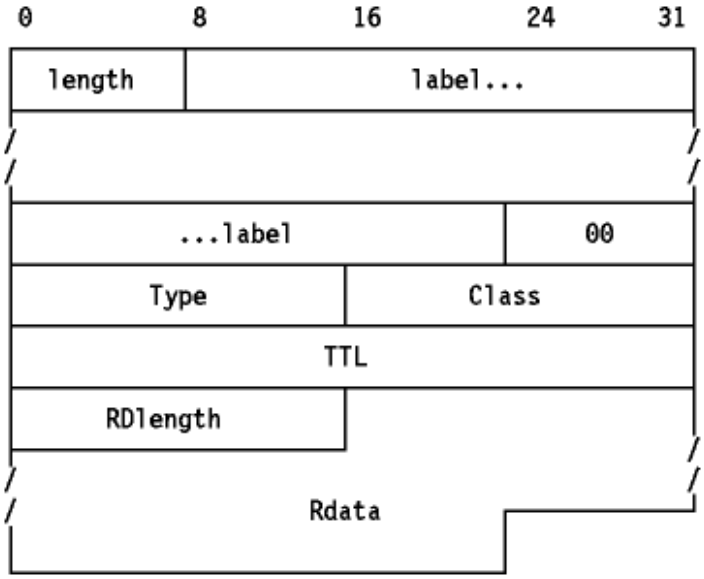


Figura: Formato de la sección "Answer" - Todos los campos están alineados por bytes. La alineación del campo "Type" a 4 bytes es un ejemplo, y no es obligatoria en el formato.

Los campos que preceden al TTL tienen el mismo significado que en la sección "question" y:

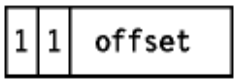
TTL
: TTL de 32-bit medido en segundos. Define cuánto tiempo se puede considerar válido el recurso.

RDlength
: Longitud de 16 bits para el campo Rdata.

Rdata
: Ristra de longitud variable cuya interpretación depende del campo "Type".

4.5.4.4 Compresión de mensajes

Con el fin de reducir el tamaño del mensaje, se utiliza un esquema de compresión para eliminar la repetición de nombres de dominio en los diversos RRs. Cualquier dominio o lista de etiquetas duplicada se sustituye por un puntero a la ocurrencia anterior. El puntero tiene la forma de un campo de 2 bytes:



- Los primeros 2 bits distinguen al puntero de una etiqueta normal, que está restringida a una longitud de 63 bytes además de el byte de longitud(con el valor de <64).
- El campo de "offset especifica un desplazamiento desde el comienzo el mensaje. Un "offset" de cero especifica el primer byte del campo ID de la cabecera.
- Si se usa compresión en un campo en el campo "Rdata" de una sección "answer", "authority" o "additional", el campo "RDlength" precedente contiene, después de haber efectuado la compresión, la longitud real .

4.5.7 Transporte

Los mensajes DNS se transmiten como datagramas(UDP) o sobre un canal(TCP).

UDP

Puerto del servidor: 53 (decimal).

Los mensajes transportados por UDP se restringen a 512 bytes. Los mensajes más largos se truncan y el bit TC de la cabecera se activa. Ya que las tramas UDP se pueden perder, hace falta una estrategia de retransmisión.

TCP

Puerto del servidor: 53 (decimal).

En este caso, el mensaje va precedido de un campo de 2 bytes que indica la longitud total de la trama.

El *STD 3 - Requerimientos del host* requiere que:

- Un "resolver" del DNS o un servidor que envía una consulta que no supone una transferencia de zona *debe* enviar una consulta UDP primero. Si la sección "answer" de la respuesta está truncada y el solicitante soporta TCP, debería intentarlo de nuevo usando TCP. Se prefiere UDP a TCP para las consultas porque UDP tiene un factor de carga mucho menor, y su uso es esencial para un servidor fuertemente cargado. El truncamiento de mensajes no suele ser un problema dados los contenidos actuales de la base de datos del DNS, ya que típicamente se pueden enviar en un datagrama 15 registros, pero esto podría cambiar a medida que se añaden nuevos tipos de registro al DNS.
- TCP debe usarse para actividades de transferencia de zonas debido a que el límite de 512 bytes de UDP siempre será inadecuado para una transferencia de zona.
- Los servidores de nombres deben soportar ambos tipos de transporte.

4.5.8 Referencias

Los siguientes RFCs definen el estándar DNS y la información el mismo:

- *RFC 1032 - Guía del administrador de dominios*
- *RFC 1033 - Guía operativa del administrador de dominios*
- *RFC 1034 - Nombres de dominio - Conceptos y servicios*
- *RFC 1035 - Nombres de dominio - Implementación y especificación*
- *RFC 1101 - Codificación DNS de nombres de red y de otros tipos*
- *RFC 1183 - Nuevas definiciones de RRs de DNS*
- *RFC 1706 - Registros de recursos DNS NSAP*

4.5.9 Aplicaciones de DNS

Muchas implementaciones de DNS proporcionan tres utilidades bastante comunes para consultar a servidores de nombres:

host
: Obtiene una dirección IP asociada con un nombre de host o un nombre de host asociado con una dirección IP.

nslookup
: Permite localizar información acerca de los nodos de red, examinar los contenidos de la base de datos de un servidor de nombres y establecer la accesibilidad a servidores de nombres.

dig("Domain Internet Groper")
: Permite probar los servidores de nombres, reunir grandes volúmenes de información de nombres de dominio y ejecutar simples consultas de nombres de dominio.

 Tabla de contenidos  SMTP ("Simple Mail Transfer Protocol")

4.6 SMTP("Simple Mail Transfer Protocol")

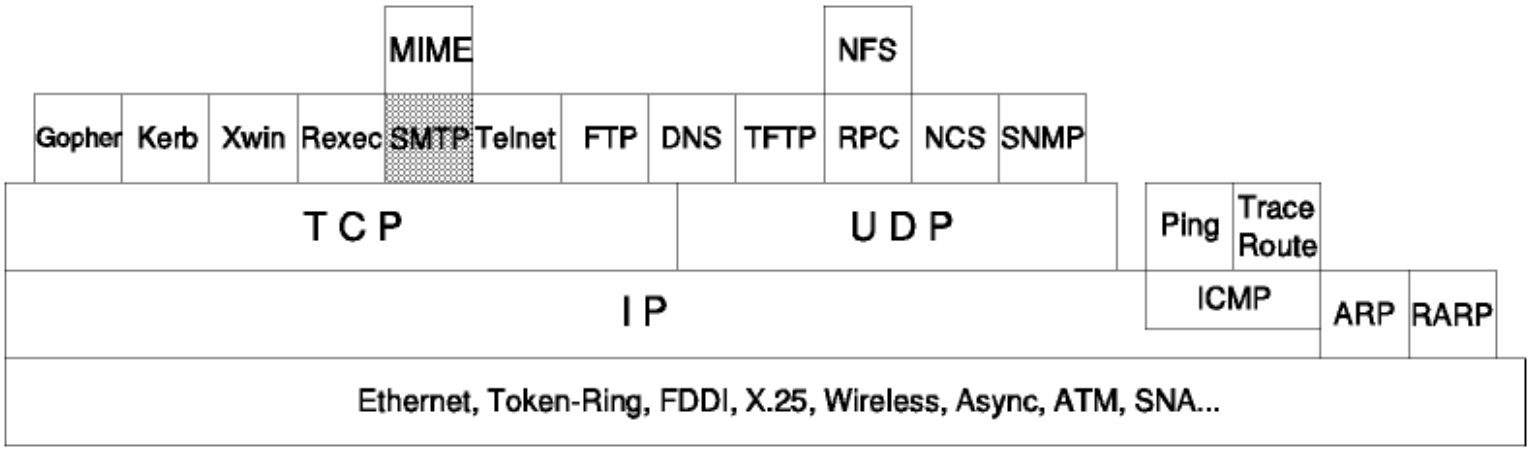


Figura: SMTP("Simple Mail Transfer Protocol")

El correo electrónico (E-mail) es probablemente la aplicación TCP/IP más usada. Los protocolos de correo básicos de correo proporcionan intercambio de correo y mensajes entre hosts TCP/IP hosts; se han añadido servicios para la transmisión de datos que no se pueden representar con texto ASCII de 7 bits.

Hay tres *protocolos estándares* que se aplican a este tipo de correo. Todos son *recomendados*. El término SMTP se emplea con frecuencia para referirse a la combinación de los tres protocolos, por su estrecha interrelación, pero estrictamente hablando, SMTP es sólo uno de los tres. Normalmente, el contexto hace evidente de cuál de los tres se está hablando. Cuando haya ambigüedad, se emplearán los números STD o RFC. Los tres estándares son:

- Un estándar para el intercambio de correo entre dos ordenadores(STD 10/RFC 821), que especifica el protocolo usado para enviar correo entre hosts TCP/IP. Este estándar es SMTP.
- Un estándar(STD 11) para el formato de los mensajes de correo, contenido en dos RFCs. El RFC 822 describe la sintaxis de las cabeceras y su interpretación. El RFC 1049 describe como un conjunto de documentos de tipos diferentes del texto ASCII plano se pueden usar en el cuerpo del correo(los mismos documentos están en ASCII de 7 bits con información de formato embebida: Postscript, Scribe, SGML, TEX, TROFF y DVI aparecen en el estándar).

  El nombre oficial del protocolo para este estándar es MAIL.

- Un estándar para el encaminamiento de correo usando el DNS, descrito en el RFC 974. El nombre oficial del protocolo para este estándar es DNS-MX.

El STD 10/RFC 821 establece que los datos enviados por SMTP son ASCII de 7-bis, con el bit de orden superior a cero. Esto es adecuado para mensajes en inglés, pero no para otros lenguajes o datos que no sean texto. Hay dos estrategias para superar estas limitaciones:

- MIME("Multipurpose Internet Mail Extensions"), definido en los RFCs 1521 y 1522, que especifica un mecanismo para codificar texto y datos binarios en ASCII de 7 bits en el mensaje RFC 822. MIME se describe en MIME("Multipurpose Internet Mail Extensions").
- SMTPSE("SMTP Service Extensions"), que define un mecanismo para extender las posibilidades de SMTP más allá de las limitaciones impuestas por RFC 821. Actualmente hay tres RFCs que lo describen:
  - Un estándar para que un receptor SMTP informe al emisor que extensiones de servicio soporta(SMTPSE) soporta (RFC 1651).

    El RFC 1651 modifica el 821 para permitir que un cliente agente SMTP solicite al servidor una lista de las extensiones de servicio que soporta el inicio de una sesión SMTP. Si el servidor no soporta este RFC, responderá con un error y el cliente podrá terminar la sesión o intentar iniciar una sesión según las reglas RFC 821. Si sí lo soporta, puede responder con una lista de las extensiones que soporta. IANA mantiene un registro de servicios: la lista inicial del RFC 1651 contiene los comandos listados en el *RFC 1123 - Requerimientos para hosts de Internet - Aplicación y soporte* como opcionales en servidores SMTP. Se han definido otras extensiones con RFCs del modo habitual. Los dos siguientes RFCs definen extensiones específicas:

  - Un protocolo para transmisión de texto de 8 bits(RFC 1652) que permite a un servidor SMTP indicar que puede aceptar datos formados por bytes de 8 bits. Un servidor que informa que dispone de esta extensión no debe modificar el bit de orden superior de los bytes recibidos en un mensaje SMTP si el cliente así se lo pide.

    Las extensiones de MIME y SMTP son estrategias que se complementan más que competir entre sí. En particular, el RFC 1652 se titula *SMTPSE para transporte MIME en codificación "8bit"*, ya que MIME permite declarar mensajes con bytes de 8 bits, en vez de 7. Tales mensajes no se pueden transmitir con agentes SMTP que sigan estrictamente el RFC 821, pero se pueden transmitir cuando tanto el cliente como el servidor siguen los RFCs 1651 y 1652. Siempre que un cliente intenta enviar datos de 8 bits a un servidor que no soporta esta extensión, el cliente SMTP debe codificar el mensaje a 7 bits según el estándar MIME o devolver un mensaje de error permanente al usuario.

    Esta extensión no permite el envío de datos binarios arbitrarios porque el RFC 821 fija la longitud máxima de las líneas aceptadas por un servidor SMTP a 1000 caracteres. Los datos que no son texto pueden tener con facilidad secuencias de más de 1000 caracteres sin una secuencia <CRLF>.

    Nota: Las extensiones limitan específicamente el uso de caracteres no ASCII(aquellos con valor decimal superior a 127) al cuerpo de los mensajes - *no* están permitidos en las cabeceras RFC 822.

  - Un protocolo para la declaración del tamaño del mensaje(RFC 1653) que permite a un servidor informar al cliente del tamaño máximo de mensaje que puede aceptar. Sin esta extensión, un cliente sólo puede ser informado de que un mensaje ha excedido el tamaño máximo(sea fijo o temporal, por falta de espacio en el servidor) tras transmitir todo el mensaje. Cuando esto sucede, el servidor desecha el mensaje. Con ella, el cliente puede declarar el tamaño estimado del mensaje y el servidor devolverá un error si es demasiado grande.

  Todas estas extensiones son *borradores* y tienen status *electivo*.

4.6.1 Cómo funciona SMTP

SMTP está basado en *la entrega punto-a-punto*; un cliente SMTP contactará con el servidor SMTP del host de destino directamente para entregar el correo. Guardará el correo hasta que se haya copiado con éxito en el receptor. Esto difiere del principio de retransmisión común a muchos sistemas de correo en las que el correo atraviesa un número de host intermedios de la misma red y donde una transmisión con éxito implica sólo que el correo ha alcanzado el host correspondiente al siguiente salto.

En varias implementaciones, existe la posibilidad de intercambiar correo entre los sistemas de correo locales y SMTP. Estas aplicaciones se denominan *pasarelas o puentes de correo*. Enviar correo a través de una pasarela puede alterar la entrega punto-a-punto, ya que SMTP sólo garantiza la entrega fiable a la pasarela, no al host de destino, más allá de la red local. La transmisión punto SMTP en estos casos es host-pasarela, pasarela-host o pasarela-pasarela; SMTP no define lo que ocurre más allá de la pasarela. CSNET proporciona un interesante ejemplo de servicio de pasarela de correo. Diseñada en principio como un servicio barato para interconectar centros científicos y de investigación, CSNET opera una pasarela que permite a sus suscriptores enviar y recibir correo en Internet con sólo un módem con dial. La pasarela sondea a los suscriptores a intervalos regulares, les entrega su correo y recoge el correo de salida. A pesar de no ser una entrega punto-a-punto, ha demostrado ser un sistema muy útil.

Cada mensaje tiene:

- Una cabecera, o sobre, con estructura RFC 822. La cabecera termina con una línea nula(una línea con sólo la secuencia <CRLF>).
- Contents

    Todo lo que hay tras la línea nula es el cuerpo del mensaje, una secuencia de líneas con caracteres ASCII(aquellos con valor menor del 128 decimal).

El RFC 821 define un protocolo cliente/servidor. Como siempre, el cliente SMTP es el que inicia la sesión (el emisor) y el servidor el que responde a la solicitud de sesión(el receptor). Sin embargo, como el cliente suele actuar como servidor para un programa de correo del usuario, es más sencillo referirse a él como emisor SMTP, y al servidor como receptor SMTP.

4.6.1.1 Formato de la cabecera

Normalmente, el usuario no tiene por qué preocuparse de la cabecera, que es responsabilidad de SMTP.

El RFC 822 contiene un análisis completo de la cabecera. La sintaxis es BNF("Backus-Naur Form") extendida. El RFC 822 contiene una descripción de BNF, y muchos RFCs relacionados usan el mismo formato. Además describe como convertir una cabecera a su *forma canónica*, uniendo las líneas de continuación, los espacios no significativos, los comentarios, etc. Es una sintaxis poderosa, pero relativamente difícil de analizar. Aquí se incluye una breve descripción.

Brevemente, la cabecera es una lista de líneas de la forma:

```
field-name: field-value
```

Los campos comienzan en la columna 1: las líneas que comienzan con caracteres en blanco(SPACE o TAB) son líneas de continuación que se unen para crear una sola línea para cada campo en la forma canónica. Las cadenas entre comillas ASCII señalan que los caracteres especiales que limitan no son significativos sintácticamente. Muchos valores importantes(como los de los campos "To" y "From") son buzones. Las formas más corrientes para estos son:

- octopus@garden.under.the.sea
- The Octopus <octopus@garden.under.the.sea>
- "The Octopus" <octopus@garden.under.the.sea>

La cadena "The Octopus" ha de ser leída por receptores humanos y es el nombre del propietario del buzón. "octopus@garden.under.the.sea" es la dirección para la máquina del buzón(el > y el < delimitan la dirección pero no forman parte de ella). Se ve que esta forma de direccionamiento está relacionada con DNS. De hecho, el cliente SMTP utiliza el DNS para determinar la dirección de destino del buzón.

Algunos campos habituales son:

keyword

---

valor

to
: Receptores primarios del mensaje.

cc
: Receptores Secundario("carbon-copy") del mensaje.

from
: Identidad del emisor.

reply-to
: El buzón al que se han de enviar las repuestas. Este campo lo añade el emisor.

return-path
: Dirección y ruta hasta el emisor. Lo añade el sistema de transporte final que entrega el correo.

Subject
: Resumen del mensaje. Suele proporcionarlo el usuario.

4.6.1.2 Intercambio de correo

El diseño de SMTP se basa en el modelo de comunicación mostrado en Figura - Modelo para SMTP. Como resultado de la solicitud de correo de un usuario, el emisor SMTP establece una conexión en los dos sentidos con el receptor SMTP. El receptor puede ser el destinatario final o un intermediario(pasarela de correo). El emisor generará comandos a los que replicará el receptor.

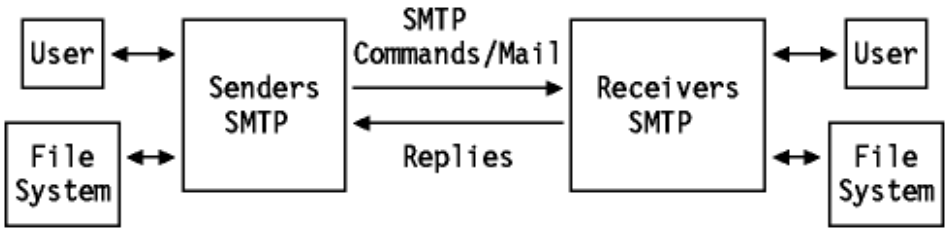


Figura: Modelo para SMTP

Flujo de transacción de correo de SMTP:

Aunque los comandos y réplicas de correo están definidas rígidamente, el intercambio se puede seguir en Figura - Flujo de datos normal de SMTP. Todos los comandos, réplicas o datos intercambiados son líneas de texto, delimitadas por un <CRLF>. Todas las réplicas tienen un código numérico el comienzo de la línea.

1. El emisor SMTP establece una conexión TCP con el SMTP de destino y espera a que el servidor envíe un mensaje *"220 Service ready"* o *"421 Service not available"* cuando el destinatario es temporalmente incapaz de responder.
2. Se envía un HELO (abreviatura de "hello"), con el que el receptor se identificará devolviendo su nombre de dominio. El SMTP emisor puede usarlo para verificar si contactó con el SMTP de destino correcto.

   Si el emisor SMTP soporta las extensiones de SMTP definidas en el RFC 1651, puede sustituir el comando HELO por EHLO. Un receptor SMTP que no soporte las extensiones responderá con un mensaje *"500 Syntax error, command unrecognized"*. El emisor SMTP debería intentarlo de nuevo con HELO, o si no puede retransmitir el mensaje sin extensiones, enviar un mensaje QUIT.

   Si un receptor SMTP soporta las extensiones de servicio, responde con un mensaje multi-línea *250 OK* que incluye una lista de las extensiones de servicio que soporta.

3. El emisor inicia ahora una transacción enviando el comando MAIL al servidor. Este comando contiene la ruta de vuelta al emisor que se puede emplear para informar de errores. Nótese que una ruta puede ser más que el par *buzób@nombre de dominio del host*. Además, puede contener una lista de los hosts de encaminamiento. Si se acepta, el receptor replica con un *"250 OK"*.
4. El segundo paso del intercambio real de correo consiste en darle al servidor SMTP el destino del mensaje(puede haber más de un receptor). Esto se hace enviando uno o más comandos RCPT TO:<*forward-path*>. Cada uno de ellos recibirá una respuesta *"250 OK"* si el servidor conoce el destino, o un *"550 No such user here"* si no.
5. Cuando se envían todos los comandos rcpt, el emisor envía un comando *DATA* para notificar al receptor que a continuación se envían los contenidos del mensaje. El servidor replica con *"354 Start mail input, end with <CRLF>.<CRLF>"*. Nótese que se trata de la secuencia de terminación que el emisor debería usar para terminar los datos del mensaje.
6. El cliente envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF> que el servidor reconoce con *"250 OK"* o el mensaje de error apropiado si cualquier cosa fue mal.
7. Ahora hay varias acciones posibles:
   - El emisor no tiene más mensajes que enviar; cerrará la conexión con un comando QUIT, que será respondido con *"221 Service closing transmission channel"*.
   - El emisor no tiene más mensajes que enviar, pero está preparado para recibir mensajes(si los hay) del otro extremo. Mandará el comando TURN. Los dos SMTPs intercambian sus papelees y el emisor que era antes receptor puede enviar ahora mensajes empezando por el paso 3 de arriba.
   - El emisor tiene otro mensaje que enviar, y simplemente vuelve al paso 3 para enviar un nuevo MAIL.

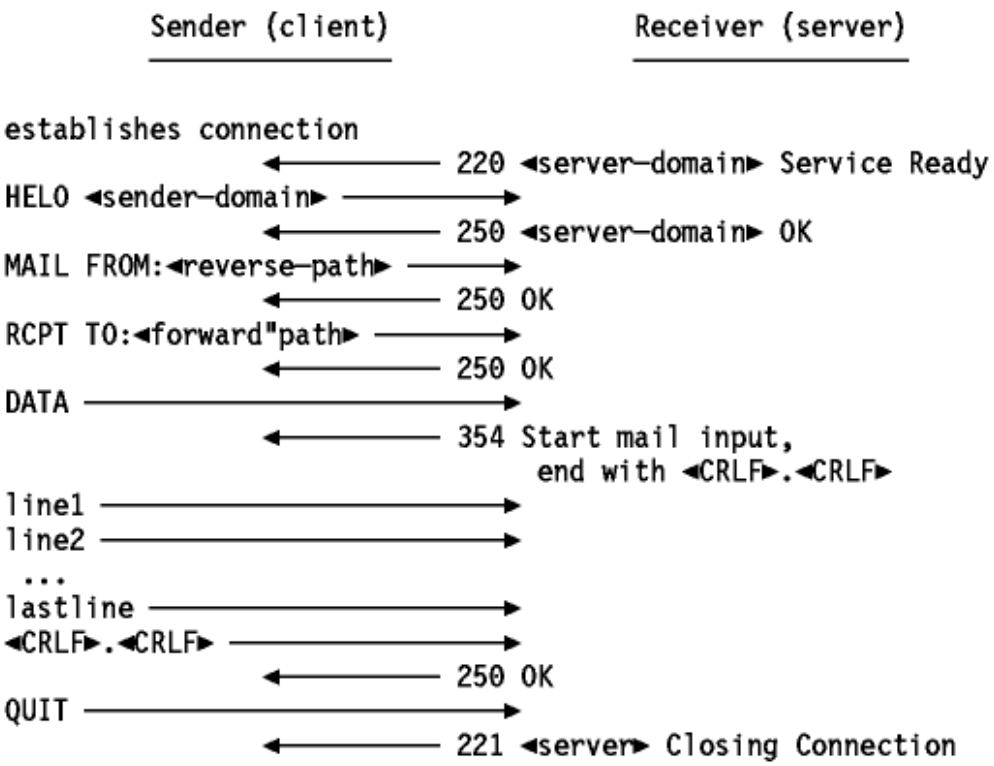


Figura: Flujo de datos normal de SMTP - Se entrega un correo al buzón de destino.

La dirección de destino SMTP(dirección de buzón),

en su forma general, *parte-localt@nombre de dominio*, puede adoptar distintos esquemas:

usuario@host
: Para un destino directo en la misma red TCP/IP.

usuario%host.remoto@host-pasarela
: Para un usuario en un host remoto de destino no SMTP, vía una pasarela.

@host-a,@host-b:usuario@host-c
: Para un mensaje *retransmitido*. Contiene explícitamente información de encaminamiento. El mensaje será entregado primero al host a, que lo retransmitirá al host b. El host b enviará el mensaje al host de destino real, el c. Nótese que el mensaje se almacena en cada uno de los host intermedios, por lo que no se necesita un mecanismo de entrega punto-a-punto.

En la descripción anterior, sólo los comandos más importante se han mencionado. Todos ellos son comandos que deben estar reconocidos en cualquier implementación SMTP. Existen otros comandos, pero la mayoría son opcionales, es decir, el RFC no los requiere. Sin embargo, implementan funciones muy interesantes tales como retransmisión, correo, listas, etc.

Para una lista completa de comandos, ver el *RFC 821 - SMTP("Simple Mail Transfer Protocol")* y el *RFC 1123 - Requerimientos para hosts de Internet - Aplicación y soporte*. Para detalles de las extensiones SMTP, ver los RFCs *1651 - SMTPSE("SMTP Service Extensions")*, *1652 - SMTPSE para transporte MIME con codificación "8bit"* y *1653 - SMTPSE para la declaración de tamaños de mensaje*.

Ejemplo:

En el siguiente escenario, el usuario abc en el host vm1.stockholm.ibm.comando envía una nota a los usuario xyz, opq u rst en el host delta.aus.edu. Las líneas precedidas por R: son las enviadas por el receptor, las que empiezan por S: las enviadas por el emisor.

```
 R: 220 delta.aus.edu Simple Mail Transfer Service Ready
 S: HELO stockholm.ibm.comando
 R: 250 delta.aus.edu

 S: MAIL FROM:<abc@stockholm.ibm.comando>
 R: 250 OK

 S: RCPT TO:<xyz@delta.aus.edu>
 R: 250 OK
 S: RCPT TO:<opq@delta.aus.edu>
 R: 550 No such user here
 S: RCPT TO:<rst@delta.aus.edu>
 R: 250 OK

 S: DATA
 R: 354 Start mail input, end with <CRLF>.<CRLF>
 S: Date: 23 Jan 89  18:05:23
 S: From: Alex B. Carver <abc@stockholm.ibm.comando>
 S: Subject: Important meeting
 S: To:  <xyz@delta.aus.edu>
 S: To:  <opq@delta.aus.edu>
 S: cc:  <rst@delta.aus.edu>
 S:
 S: Blah blah blah
 S: etc.....
 S: .
 R: 250 OK

 S: QUIT
 R: 221 delta.aus.edu Service closing transmission channel
```

Nótese que la cabecera del mensaje es parte de los datos a transmitir.

4.6.2 SMTP y el DNS

Si la red usa el concepto de dominio, un SMTP no puede entregar simplemente correo a TEST.IBM.comando abriendo una conexión TCP con TEST.IBM.comando. Primero debe consultar al servidor de nombres para hallar a que host(en un nombre de dominio) debería entregar el mensaje.

Para la entrega de mensajes, el servidor de nombres almacena los RRs("resource records") denominados MX RRs. Mapean un nombre de dominio a dos valores:

- Un valor de preferencia. Como pueden existir múltiples RRs MX para el mismo nombre de dominio, se les asigna una prioridad. El valor de prioridad más bajo corresponde al registro de mayor preferencia. Esto es útil siempre que el host de mayor preferencia sea inalcanzable; el emisor SMTP intenta conectar con el siguiente host en orden de prioridad.
- Un nombre de host.

También es posible que el servidor de nombres responda con una lista vacía de RRs MX. Esto significa que el nombre de dominio se halla bajo la autoridad del servidor, pero no tiene ningún MX asignado. En este caso, el emisor SMTP puede intentar establecer la conexión con el mismo nombre del host.

El RFC 974 da una recomendación importante. Recomienda que tras obtener los registros MX, el emisor SMTP debería consultar los registros WKS(*Well-Known Services*) del host, y chequear que el host referenciado tiene como entrada WKS a SMTP.

Nota: Esto es sólo una opción del protocolo, aunque aparece en numerosas implementaciones.

Aquí hay un ejemplo de RRs MX:

```
fsc5.stn.mlv.fr.        IN    MX 0  fsc5.stn.mlv.fr.
                        IN    MX 2  psfred.stn.mlv.fr.
                        IN    MX 4  mvs.stn.mlv.fr.
                        IN    WKS   152.9.250.150 TCP (SMTP)
```

En el ejemplo anterior, el correo para fsc5.stn.mlv.fr debería, por prioridad, ser entregado al propio host, pero en caso de que el host sea inalcanzable, el correo también podría ser entregado a psfred.stn.mlv.fr o a mvs.stn.mlv.fr (si psfred.stn.mlv.fr no se pudiera alcanzar tampoco).

4.6.3 Servidores de correo POP("Post Office Protocol")

Debido a que un receptor de correo SMTP es un servidor, y SMTP es una aplicación punto-a-punto más que de retransmisión, es necesario que el servidor esté disponible cuando un cliente desea enviarle correo. Si el servidor SMTP reside en una estación de trabajo o en un PC, ese host debe estar ejecutando el cuando el cliente quiera transmitir. Esto no suele ser un problema en sistemas multiusuario porque están disponibles la mayor parte del tiempo. En sistemas monousuario, sin embargo, este no es el caso, y se requiere un método para asegurar que el usuario tiene un buzón disponible en otro servidor. Hay varias razones por las que es deseable descargar a la estación de trabajo de las funciones del servidor de correo, entre ellas la falta de recursos en estaciones de trabajo pequeñas, la falta o encarecimiento de la conectividad TCP, etc.

La estrategia más simple es, por supuesto, usar un sistema multiusuario para las funciones de correo, pero esto no suele ser deseable -- quizá el usuario no lo va a usar para nada más, o quiere tener acceso a Alternativamente, el usuario final puede ejecutar un cliente que comunique con un programa servidor en un host. Este servidor actúa tanto como emisor como receptor. Recibe y envía el correo del usuario.

Un método intermedio es descargar la función de servidor SMTP de la estación de trabajo del usuario final, pero no la función de cliente. Es decir, el usuario envía correo directamente desde la estación, pero tiene un buzón en un servidor. El usuario debe conectar con el servidor para recoger su correo.

El POP describe cómo un programa que se ejecuta en una estación de trabajo final puede recibir correo almacenado en sistema servidor de correo. POP usa el término "maildrop" para referirse a un buzón gestionado por un servidor POP. POP 3 es un borrador y su status es *elective*. Las versión es un *protocolo histórico* con status *no recomendado*.

4.6.3.1 Direccionando buzones en servidores

Cuando un usuario emplea un servidor para las funciones de correo, la dirección del buzón que ven otros usuarios SMTP se refiere exclusivamente al servidor. Por ejemplo, si dos sistemas se llaman:

- hayes.itso.ral.ibm.comando y
- itso180.itso.ral.ibm.comando

usándose el primero como cliente y el segundo como servidor, la dirección de correo podría ser:

- hayes@itso180.itso.ral.ibm.comando

Esta dirección de buzón aparecería en el campo "From:" de la cabecera de todo el correo saliente y en los comandos SMTP a servidores remotos lanzados por el servidor.

Sin embargo, cuando el usuarios emplea un servidor POP, la dirección de correo contiene el nombre de host de la estación de trabajo(por ejemplo steve@hayes.itso.ral.ibm.comando). En este caso, el emisor debería incluir un campo "Reply-To:" en la cabecera para indicar que las réplicas *no* se deberían enviar al emisor. Por ejemplo, la cabecera podría tener este aspecto:

```
Date: Fri, 10 Feb 95 15:38:23
From: steve@hayes.itso.ral.ibm.comando
To: "Steve Hayes" <tsgsh@gford1.warwick.uk.ibm.comando>
Reply-To: hayes@itso180.itso.ral.ibm.comando
Subject: Test Reply-To: header field
```

Se espera que el agente de correo envíe las respuestas a la dirección "Reply-To:" y no a "From:".

Usando DNS para dirigir correo

Una alternativa al uso del campo "Reply-To:" es usar el DNS para dirigir el correo al buzón correcto. El administrador del DNS con autoridad para el dominio que contiene la estación del usuario y el servidor de nombres pueen añadir registros MX al DNS para dirigir el correo, tal como se describe en SMTP y el DNS. Por ejemplo, los siguientes registros MX indican a los clientes SMTP que, si el servidor SMTP en hayes.itso.ral.ibm.comando no está disponible, hay un servidor de correo en itso.180.ral.ibm.comando (9.24.104.180) que se debería usar en su lugar.

```
itso180.itso.ral.ibm.comando.  IN     WKS  9.24.104.180 TCP (SMTP)

hayes.itso.ral.ibm.comando.    IN     MX 0 hayes.itso.ral.ibm.comando.
                           IN     MX 1 itso180.itso.ral.ibm.comando.
```

4.6.4 Pasarelas SMTP

Una pasarela SMTP es un host con dos conexiones a redes distintas. Las pasarelas SMTP se pueden implementar de forma que conecten distintos tipos de redes.

Una pasarela SMTP-RSCS/NJE se configura utilizando un fichero de configuración SMTP como el que se muestra abajo. Para configurar un host que no es pasarela, no se debe especificar la sentencia GATEWAY.

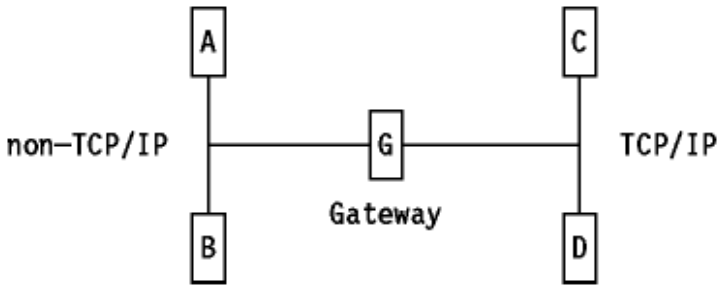


Figure: SMTP-RSCS/NJE Mail Gateway

```
 .........
;
GATEWAY                ; accept mail from and deliver mail to RSCS host
RSCSDOMAIN RSCSNET     ; pseudo domain name of associated RSCS network
```

```
LOCALFORMAT NETDATA    ; local recipients receive mail in Netdata format
RSCSFORMAT NETDATA     ; RSCS recipients receive mail in Netdata format
REWRITE822HEADER NO    ; Only set to no if you do not want SMTP to
                       ; rewrite the 822 headers on all mail passing
                       ; from RSCS to TCP through gateway.
```

Se puede prohibir el acceso a la pasarela a determinados nodos de la red, empleando la sentencia de configuración RESTRICT.

Alternativamente, la seguridad se puede implementar con un fichero de autorización de accesos, que es una tabla en la que se especifican de quién y a quién se puede enviar correo por la pasarela. La siguiente tabla es un ejemplo de este tipo de archivo:

```
*
*
*
  DEBULOI   MLVFSC0
  DEBULOIS  MLVFSC1       FRED0        Y               N
  DEBULOIS  MLVFSC5       FRED1        N               Y
  TCPMAINT  MLVFSC5       TCP0         N               N
  DEBULOIS  MLVFSC1       TCP1         Y               Y
```

Los sistemas IBM VM y VMS están preparados para funcionar como pasarelas de correo seguras. Así mismo, OS/400 se puede configurar para que funcione como pasarela SMTP

4.6.5 Referencias

Una descripción detallada de los estándares SMTP, MAIL y DNS-MX se puede hallar en los siguientes RFCs:

- *RFC 821 - SMTP("Simple Mail Transfer Protocol")*
- *RFC 822 - estándar para el formato de los mensajes de texto para la red ARPA*
- *RFC 974 - Encaminamiento de correo y el DNS*
- *RFC 1049 - Un campo "Content Type" para la cabecera de los mensajes de Internet*

Se puede encontrar una detallada descripción del POP en los siguientes RFCs:

- *RFC 937 - POP("Post Office Protocol") - Versión 2*
- *RFC 1725 - POP("Post Office Protocol") - Versión 3*

Tabla de contenidos MIME ("Multipurpose Internet Mail Extensions")

4.7 MIME("Multipurpose Internet Mail Extensions")

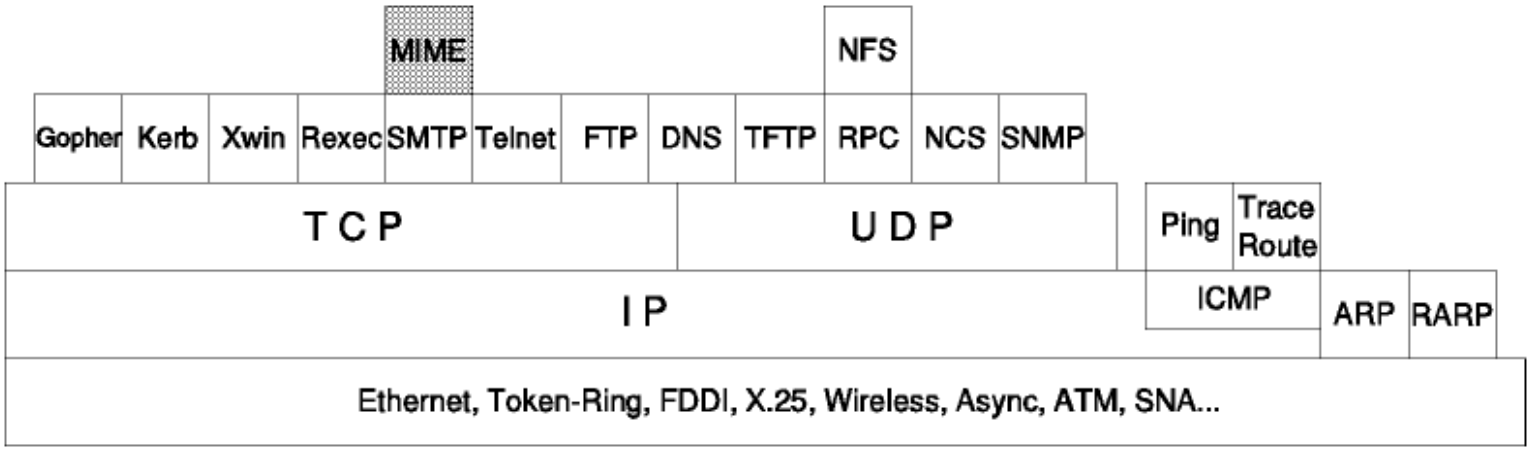


Figura: MIME("Multipurpose Internet Mail Extensions")

El MIME es un *borrador*. Su status es *electivo*.

El E-mail (como se describe en SMTP("Simple Mail Transfer Protocol")) es probablemente la aplicación TCP/IP más usada. Sin embargo, SMTP (es decir, un sistema de correo que sigue la norma STD 10/RFC 821) se limita a texto ASCII de 7 bits con una longitud de línea máxima de 100 caracteres lo que da lugar a diversas limitaciones.

- SMTP no puede transmitir ficheros ejecutables u otros objetos binarios. Hay un número de métodos ad hoc para encapsular objetos binarios en correos SMTP, por ejemplo:
  - Codificar el fichero en formato hexadecimal puro.
  - Las utilidades UNIX UUencode y UUdecode usadas para codificar datos binarios en el sistema de correo UUCP para superar las limitaciones del formato de 7 bits.
  - La representación Andrew Toolkit.

  Ninguna de ellas se puede considerar un estándar de facto. UUencode es quizás la más extendida debido a que los sistemas UNIX son los pioneros de Internet.

- SMTP no puede transmitir texto que incluya caracteres nacionales, ya que estos se representan con valores iguales o mayores de 128 en los juegos de caracteres basados en ASCII.
- Los servidores SMTP pueden rechazar los correos mayores de un tamaño concreto. Cualquier servidor puede tener límites permanentes y/o temporales a la máxima cantidad de correo que puede aceptar de un cliente en un momento dado.
- Las pasarelas SMTP que traducen de ASCII a EBCDIC y viceversa no emplean un conjunto consistente de mapeados de páginas de código, dando lugar a problemas de traducción.
- as pasarelas SMTP a redes X.400 no pueden manejar datos que no sean texto en los correos de X.400. Los estándares para mapear X.400 al STD 11/RFC 822(16) anterior al MIME requieren que las partes que no sean texto del cuerpo de un mensaje X.400 se conviertan o se desechen y el receptor RFC 822 informe que se ha desechado esa información.

  Obviamente, esto no es deseable ya que se trata de información presumiblemente importante aunque el sistema de correo no la pueda entender. Desecharla significa que será inaccesible al usuario. Convertirla a ASCII implica poner a cero el byte de orden superior, lo que supone la corrupción de los datos más haya de cualquier forma de recuperación. El problema es particularmente agudo en el caso de que emisor y receptor se hallen en redes X.400 pero el correo pase por una red RFC 822 intermedia("tunnelling") ya que los usuarios de X.400 esperan recibir correos X.400 sin perder información.

- Algunas implementaciones de SMTP u otros MTAs("Mail Transport Agents") de Internet no se adhieren por completo al estándar SMTP definido en el RFC 821. Problemas habituales son:
  - Eliminación de espacios en blanco terminales(TABs y SPACEs)
  - Relleno de todas las líneas de un mensaje para que tengan la misma longitud.
  - Empaquetado de las líneas de más de 76 caracteres.
  - Cambio de secuencias de nueva línea para diferentes convenios(por ejemplo, los caracteres <CR> se pueden convertir a secuencias<CRLF>).
  - Conversión de caracteres TAB a múltiples caracteres SPACE.

MIME es un estándar que incluye mecanismos para resolver estos problemas en una forma con un alto grado de compatibilidad con los estándares RFC 822. Debido a que los mensajes de correo suelen enviarse a través de pasarelas de correo, a un cliente SMTP no le es posible distinguir entre un servidor que gestiona el buzón del receptor y uno que actúa como pasarela a otra red. Como el correo que atraviesa una pasarela se puede dirigir a otras pasarela, que pueden usar distintos protocolos de mensajería, en el caso general al emisor no le es posible determinar el mínimo común denominador de las capacidades de cada una de las etapas que atraviesa el mensaje. Por este motivo, MIME asume el peor caso: transporte ASCII de 7 bits que puede no seguir estrictamente el RFC 821. No define ninguna extensión al RFC 821, sino que se limita sólo a las extensiones incluidas en el RFC 822. De esta forma, un mensaje MIME puede ser enrutado a través de cualquier número de redes capaces de transmitir mensajes RFC 821. Se describe en dos partes:

- Protocolos para incluir objetos distintos de los mensajes de correo US ASCII. dentro de mensajes RFC 822. El RFC 1521 los describe.
- Un protocolo para codificar texto no US ASCII en los campos de cabecera según el RFC 822. El RFC 1522 lo describe.

Aunque el RFC 1521 proporciona un mecanismo adecuado para describir dato de texto de mensajes X.400 en una forma no compatible con el RFC 822, no dice como se han de mapear las partes de los mensajes X.400 a las de los mensajes MIME. Esta conversión está definida en los RFCs 1494, 1495 y 1496 que actualizan los protocolos de conversión de RFC 822 a X.400.

El estándar MIME se diseñó con el siguiente orden general de prioridades:

1. Compatibilidad con estándares ya existentes como el RFC 822.

   Hay dos áreas en las que la compatibilidad con estándares previos no es completa.

- El RFC 1049 (que es parte del 11) describe un campo "Content-Type:" usado para indicar el tipo de datos(texto ASCII) del cuerpo del mensaje. PostScript o SGML permitirían a un agente de correo procesar sus datos de esta forma. MIME conserva el campo, pero cambia los valores definidos para él. Ya que la respuesta correcta de un agente cuando se encuentra un valor desconocido en este campo es ignorarlo, esta incompatibilidad no debería causar problemas de consideración.
- El RFC 934 discute la encapsulación de mensajes en el contexto de la retransmisión de mensajes y define las separadores de encapsulación: líneas indicando el comienzo y el fin de un mensaje encapsulado. MIME conserva bastante compatibilidad con RFC 934, pero no incluye el mecanismo de "entrecomillado" de RFC 934 para líneas de mensajes encapsulados que de otro modo se podrían confundir con separadores.(17)

La cuestión de compatibilidad más importante es que la forma estándar de un mensaje MIME sea legible con un lector de correo RFC 821. El caso es que, precisamente, la codificación estándar de los cuerpos de los mensajes MIME es exactamente RFC 822.

1. Compatibilidad general. Como se indicó arriba, hay muchos MTAs muy extendidos en Internet que no siguen el STD 10/RFC 821. Los mecanismos de codificación especificados en el RFC 1521 están diseñados para omitir los más comunes - como aquellos que pliegan las líneas a la longitud de 76 caracteres o tienen espacios en blanco redundantes - sólo transmitiendo líneas cortas sin espacios en blanco redundantes, y para permitir la codificación segura de los datos.

****** NOTA: ****** MIME *no* requiere que los objetos de correo estén codificados -- la decisión se deja al usuario y/o programa de correo. Para datos binarios, transmitidos a través de SMTP de 7 bits, la codificación es necesaria invariablemente, pero para datos que son texto en su mayoría, puede que no haga falta.

El mecanismo preferido de codificación para datos "de texto en su mayoría" es tal que, como mínimo, es , es fiable para cualquier agente SMTP en un sistema ASCII, y como máximo, es fiable con todas las pasarelas y MTAs conocidos. La razón por la que MIME no requiere codificación total es que la codificación dificulta la legibilidad cuando MIME el correo se transmite a sistemas no MIME.

---

2. Facilidad de extensión. RFC 1521 categoriza los elementos de cuerpos de correo en siete *tipos de contenido("content-types")* que tienen *subtipos*. Las parejas tipo de contenido/subtipo tienen por su parte parámetros que describen con más precisión el objeto en cuestión. El RFC define un mecanismo para registrar nuevos valores para estos y otros campos MIME en IANA. El RFC 1590, a su vez, actualiza este mecanismo.

   Para ver la lista actual de valores MIME, consultar *STD 2 - Números asignados de Interne*. El resto del capítulo describe los valores y tipos del RFC 1521.

Una consecuencia de este enfoque es que, citando al RFC 1521, "algunos de los mecanismo[usados en MIME] pueden parecer extraños o incluso barrocos al principio. En particular, siempre se ha favorecido la compatibilidad sobre la elegancia."

Debido a que el RFC 822 define la sintaxis de las cabeceras(y deliberadamente permite la extensión del conjunto de cabeceras que describe) pero no la composición de los cuerpos, el estándar MIME es muy compatible con el RFC 822, particularmente con la parte del RFC 1521 que define la estructura de los cuerpos de los mensajes y un conjunto de cabeceras que constituyen una descripción de estos.

MIME se puede ver como un protocolo de alto nivel; como trabaja enteramente dentro de los límites de los STDs 10 y 11, no implica en absoluto a los protocolos de transporte(o inferiores) de la pila de protocolos.

4.7.1 Cómo funciona MIME

Un mensaje MIME debe contener un campo de cabecera con el siguiente texto:

```
MIME-Version: 1.0
```

Como en el caso de las cabeceras RFC 822, los nombres de los campos de la cabecera MIME no son sensibles a mayúsculas y minúsculas, pero los valores de los campos pueden serlo, según su nombre y contexto. Los valores de los campos MIME señalados abajo no lo son, a menos que se diga lo contrario.

La sintaxis general para los campos de la cabecera MIME es la misma que en RFC 822, por lo que el siguiente campo:

```
MIME-Version: 1.0 (comentario)
```

es válido ya que las frases entre paréntesis se tratan como comentarios y se ignoran.

En MIME se definen cinco campos de cabecera.

MIME-Version
: Como se indica arriba, debe tener el valor "1.0"

Content-Type
: Describe como se ha de interpretar el objeto dentro del cuerpo. El valor por defecto es "text/plain; charset=us-ascii" que indica texto de 7 bits sin formato(cuerpo de mensaje según la definición de RFC 822).

Content-Transfer-Encoding
: Describe cómo está codificado el objeto de modo que se puede incluir en el correo de forma fiable.

Content-Description
: Una descripción en texto plano del objeto del cuerpo que es útil cuando el objeto no es legible (por ejemplo, datos de audio).

Content-ID
: Un valor unívoco especificando el contenido de esta parte del mensaje.

Nota: el RFC 1521 incluye una definición del término "cuerpo" tal como se emplea arriba, además de los términos "parte del cuerpo", "entidad", y mensaje. Desafortunadamente las cuatro definiciones son "pescadillas que se muerden la cola" porque el mensaje MIME genérico puede contener recursivamente mensajes MIME hasta cierta profundidad. La forma más simple de cuerpo es el cuerpo del mensaje como lo define el RFC 822.

Los dos primeros campos se describen con detalle en las siguientes seccione.

4.7.2 El campo Content-Type

El cuerpo del mensaje se describe con un campo *Content-Type* de la forma:
Content-Type: *type/subtype ;parameter=value ;parameter=value*

Los parámetros admisibles son dependientes de "type" y "subtype". Algunos pares "type/subtype", algunos los tienen opcionales, otros obligatorios, y otros ambos. El parámetro de "subtype" *no* se puede omitir, pero sí todo el campo, y en este caso el valor por defecto es *text/plain*.

Hay siete "content-types" estándares:

text
: Sólo tiene un "subtype":

plain
: Texto sin formatear. El juego de caracteres se puede especificar con el parámetro *charset*. Se admiten los siguientes valores.

us-ascii
: El texto consiste en caracteres ASCII en el rango de 0 a 127(decimal). Este es el valor por defecto(por compatibilidad con RFC 822).

iso-8859-x spotis8859
: donde la "x" está en el rango de 1 a 9 para las distintas partes del estándar ISO-8859. El texto consiste en caracteres ISO en el rango de 0 a 255(decimal). Todos los juegos de caracteres ISO-8859 están basados en ASCII con caracteres nacionales en el rango de 128 a 255. Si el juego no contiene caracteres con valores mayores de 127, se debería usar "us-ascii", ya que así se pueden representar adecuadamente.

Se pueden añadir nuevos "subtypes" para describir otros formatos de texto(tales como formatos de procesadores de texto) que contengan información para que una aplicación mejore el aspecto del texto, siempre que el software en cuestión no tenga que interpretar el significado del texto.

multipart
: El cuerpo del mensaje contiene múltiples objetos de tipos de datos independientes. En cada caso, el cuerpo se divide en partes denominadas separadores de encapsulado. El contenido de los separadores se define con un parámetro en el campo "content-type", por ejemplo:

```
Content-Type: multipart/mixed; boundary="1995021309105517"
```

: El separador no debería aparecer dentro de ninguna de las partes del mensaje. Es sensible a mayúsculas y minúsculas y tiene de 1 a 70 caracteres elegidos de un conjunto de 75 seleccionados por su robustez en pasarelas de correo, y no puede acabar en espacio. Cada separador consiste en su valor prefijado por la secuencia <CRLF> y dos guiones(para la compatibilidad con RFC 934). El último separador que marca el fin de la última parte tiene además como sufijo de dos guiones. Dentro de cada parte hay una cabecera MIME, que como las cabeceras del correo ordinario termina en la secuencia *<CRLF><CRLF>* pero que puede estar vacía. El campo de cabecera define el contenido del mensaje encapsulado.
: Se definen cuatro subtipos:

mixed
: Las distintas partes son independientes pero se han de transmitir juntas. Se le deberían mostrar al receptor en el mismo orden en el que aparecen en el mensaje.

parallel
: Difiere del anterior sólo en que las partes no tienen orden inherente y el programa receptor de correo puede, por ejemplo, mostrarlas en paralelo.

alternative
: Las distintas partes son versiones alternativas de la misma información. Se ordenan de menor a mayor fidelidad al original, y el sistema de correo receptor debería mostrar al usuario la versión más fiel.

digest
: Es una variante de "mixed" en la que el par "type/subtype" es "message/rfc822" en vez de "text/plain". Se usa en el caso frecuente de que se transmitan juntos múltiples mensajes RFC 822 o MIME.

Un ejemplo complejo se muestra en Figura - Ejemplo de un mensaje "multipart" complejo.

```
MIME-Version: 1.0
From: Steve Hayes <steve@hayessj.bedfont.uk.ibm.com>
To:   Matthias Enders <enders@itso180.itso.ral.ibm.com>
Subject: Multipart message
Content-type: multipart/mixed; boundary="1995021309105517"

Esta sección se llama preámbulo. Va tras la cabecera pero después del primer separador. Los lectores de correo que
entiendan mensajes "multipart" deben ignorarlo.
--1995021309105517

La primera parte. No hay cabecera, por lo que se trata de "text/plain" con el juego de caracteres
"charset=us-ascii"por defecto. El <CRLF> adyacente es parte de la secuencia <CRLF><CRLF>
que termina la cabecera nula. El último es parte del siguiente separador, por lo que esta parte consiste
en cinco líneas de texto con cuatro <CRLF>s.
--1995021309105517
Content-type: text/plain; charset=us-ascii
Comments: this header explicitly states the defaults

Sólo una línea, acabada en salto de línea

--1995021309105517
Content-Type: multipart/alternative; boundary=_
Comments:     An encapsulated multipart message!

De nuevo, este preámbulo se ignora. El cuerpo "multipart" contiene una imagen estática y una imagen de vídeo
codificadas en Base64.  Ver Codificación Base64
Una característica es que el carácter "_", permitido en separadores "multipart", no aparece nunca en la
codificación Base64, por lo que se puede usar un separador muy sencillo:
--_
Content-type: text/plain


Este mensaje contiene imágenes que no se pueden visualizar en la terminal.

--_
Content-type: image/jpeg
Content-transfer-encoding: base64
Comments: This photograph is to be shown if the user's system cannot display
          MPEG videos.  Only part of the data is shown in this book because
          the reader is unlikely to be wearing MIME-compliant spectacles.

Qk1OAAAAAAAAAE4EAABAAAAAQAEAAPAAAAABAAgAAAAAAAAAAAAAAAAAAAAAAAABAAAAAQAAAAAA
```

```
AAAAAAAAAAAAAAAAAAAAAAB4VjQSAAAAAAAAgAAAkgAAAJKAAKoAAACqAIAAqpIAAMHBwQDJyckA
/9uqAKpJAAD/SQAAAG0AAFVtAACqbQAA/20AAAkAABVkgAAqiQAAP+SAAAAtgAAVbYAAKq2AAD/
<base64 data continues for another 1365 lines>
--_
Content-type: video/mpeg
Content-transfer-encoding: base64

AAABswoAeBn//+CEAAABsgAAAOgAAAG4AAAAAAAAAQAAT/////wAAAGy//8AAAEBQ/ZlIwwBGWCX
+pqMiJQDjAKywS/1NRrtXcTCLgzVQymqqHAf0sL1sMgMq4SWLCwOTYRdgyAyrhNYsLhhF3DLjAGg
BdwDXBv3yMV8/4tzrp3zsAWIGAJg1IBKTeFFI2IsgutIdfuSaAGCTsBVnWdz8afdMMAMgKgMEkPE
<base64 data continues for another 1839 lines>
--_--
fin del mensaje "multipart" anidado. Este es el epílogo. Como el preámbulo, se ignora.
--1995021309105517--
Fin del mensaje.
```

Figura: Ejemplo de un mensaje "multipart" complejo

message
: el cuerpo es un mensaje encapsulado, o parte de uno. Están definidos tres subtipos:

rfc822
: el mismo cuerpo es un mensaje encapsulado con la sintaxis de un mensaje RFC 822. Sin embargo, a diferencia de los mensajes de "alto nivel" de RFC 822, no tiene que tener el formato mínimo "From:", "To:" y una cabecera de destino al menos.
Nota: RFC822 se refiere a la sintaxis de los "sobres" de los mensajes encapsulados.

partial
: Este tipo se usa para permtir la fragmentación de correos grandes de forma análoga a la fragmentación IP. Debido a que los agentes SMTP pueden imponer límites superiores al tamaño del correo, puede que sea necesario enviarlos en fragmentos. La finalidad de este campo es que la fragmentación sea transparente al receptor. El agente del usuario receptor debería reensamblar los fragmentos para crear un nuevo mensaje con semántica idéntica a la del original. Hay tres parámetros para el campo "Content-type":

id=
: Un identificador unívoco común a todas las partes del mensaje.

number=
: El número de secuencia de esta parte; la primera parte se numera con el 1.

total=
: El número total de partes. Es opcional en todas, excepto en la última. La última parte se identifica con la igualdad entre el parámetro "number" y "total".

El mensaje original es siempre un mensaje que sigue las reglas RFC 822. La primera parte es sintácticamente equivalente a un mensaje "messagge/rfc822"(es decir, el propio cuerpo contiene cabeceras) y las partes siguientes a mensajes "text/plain". Al reconstruir el mensaje, los campos de cabecera RFC822 se toman del mensaje de alto nivel, con la excepción de aquellos campos que no se pueden copiar del mensaje interior al exterior cuando existe fragmentación(por ejemplo, el campo "Content-Type").

Nota: está permito de forma explícita fragmentar aún más un mensaje "messagge/partial". Esto permite que las pasarelas de correo fragmenten los mensajes libremente para asegurarse de que todas las partes son lo bastante pequeñas para ser transmitidas. Si este no fuera el caso, el agente de correo que realice la fragmentación tendría que conocer el mínimo tamaño máximo que los correos se encontrarían en su ruta hacia el destino.

external-body
: Este tipo contiene un puntero a un objeto que existe en algún otro sitio. Tiene la sintaxis del tipo "message/rfc822". La cabecera del mensaje de alto nivel define como se ha de acceder el objeto externo, usando el parámetro *"access-type:*"(tipo de acceso) del campo "Content-Type:" y un conjunto de campos adicionales que son específicos del access-type. La finalidad de esto es que el lector de correo sea capaz de acceder de modo síncrono al objeto externo usando el "acces type". Están definidos los siguientes "access types":

ftp
: File Transfer Protocol. Se supone que el receptor ha de proporcionar el identificaddor de usuario y el password necesarios -- por razones de seguridad, estos nunca se transmiten con el mensaje.

tftp
: Trivial File Transfer Protocol.

anon-ftp
: FTP anónimo.

local-file
: Los datos están accesibles en un fichero a través del sistema de ficheros local del receptor.

afs
: Los datos están accesibles en un fichero por el sistema de ficheros global Andrew("Andrew File System").

mail-server
: Los datos son accesibles por medio de un servidor de correo. A diferencia de los otros este acceso es necesariamente asíncrono.

Cuando el objeto externo ha sido recibido, el mensaje deseado se obtiene añadiendo el objeto a la cabecera del mensaje encapsulado dentro del cuerpo del mensaje "message/external-body". Esta cabecera encapsulada tiene que se interpretada(ha de tener un campo "Content-ID:" , y normalmentte tendrá un campo "Content-Type:"). El cuerpo encapsulado no se usa(después de todo, el mensaje real está en algún otro lado) y por tanto se denomina "cuerpo fantasma". Hay una excepción a esto: si el "access type" es "mail-server" el cuerpo fantasma contiene los comandos del servidor de correo necesarios para extraer el cuerpo real del mensaje. Esto se debe a que las sintaxis del servidor de correo varían mucho y es mucho más sencillo usar el cuerpo fantasma, que de otra forma sería redundante, que codificar una sintaxis para codificar comandos arbitrarios del servidor de correo como parámetros del campo "Content-Type:".

image
: el cuerpo contiene imágenes que requieren un display gráfico o algún otro ispositivo para mostrarlas. Inicialmente hay definidos dos subtipos:

jpeg
: la imagen está en formato JPEG, codificación JFIF.

gif
: formato GIF.

video
: el cuerpo contiene imágenmes en movimiento(posiblemente con sonido sincronizado con ellas) lo que requiere una terinal inteligente o una estación de trabajo multimedia para mostrarlas. Inicialmente sólo está definido un subtipo:

mpeg
: formato MPEG.

audio
: el cuerpo contiene imágenes que requieren altavoces y una tarjeta de sonido(o hardware similar) para reproducirlas. Inicialmente sólo está definido un subtipo:

basic
: El mínimo común formato en ausencia de cualquier estándar de facto para la codificación de audio. Específicamente, es la codificación mu-law ISDN de 8 bits y un solo canal a 8kHz.

application
: Tipo orientado a tipos queno se pueden clasificar en otras categorías, y particularmente para datos a ser procesados por un programa de aplicación antes de ser

presentados al usuario, tales como hojas de texto. También va dirigido a programas de aplicación que han de ser procesados como parte del proceso de lectura del correo(por ejemplo, el tipo PostScript indicado abajo). Este tipo de uso supone graves riesgos de seguridad a menos que la implementación asegure que los mensajes de correo ejecutables se ejecuten en un entorno seguro o de "celda acolchada".
Inicialmente hay dos subtipos definidos:

PostScript
: Adobe Systems PostScript (nivel 1 o 2).
Cuestiones de seguridad: aunque suele pensarse que PostScript es un formato de impresión, es un lenguaje de programación y el uso de un intérprete de PostScript para procesar el tipo "application/PostScript" es una amenaza potencial a la seguridad. Cualquier lector de correo que interprete automáticamente programas de PostScripts es equivalente, en principio, a uno que ejecute automáticamente los programas ejecutables que recibe. El RFC 1521 perfila las posibles consecuencias.

octet-stream
: Este subtipo indica datos binarios generales consistentes en bytes de 8 bits. Es además el subtipo que un lector de correo debería asumir al encontrarse un tipo o subtipo desconocidos. Se permite cualqier parámetro, y el RFC menciona dos: *"type="*, para informar al receptor del tipo general y *"padding="* para indicar un flujo de bits codificado en un flujo de bytes(su valor es el número de ceros añadidos para alinear el flujo a un número entero de bytes).
Se recomienda que las implementaciones ofrezcan al usuario la opción de utilizar los datos como entrada a un programa de usuario o de almacenarlos en un fichero(no hay estándar para el nombre por defecto de tal fichero, aunque el RFC 1521 menciona un campo "Content-Disposition:" a ser definido en un RFC posterior.
Cuestiones de seguridad: el RFC desaconseja enérgicamente que una implementación ejecute una parte "application/octet-stream" automáticamente o que la emplee como entrada a un programa especificado en la cabecera. Hacerlo implicaría exponer la integridad del sistema y de cualquier red a la que esté conectado.

Obviamente, hay muchos tipos de datos que no caen dentro de ninguno de los subtipos indicados arriba. Los programas de correo cooperativos, manteniendo las reglas del RFC 822, usan tipos y/o subtipos que comienzan por "X-" como valores privados. No se permiten otros valores que no hayan sido registrados previamente con IANA. Ver el RFC 1590 para más detalles. La intención es que se necesiten pocos o ningún tipo adicional, pero que se añadan muchos subtipo al conjunto.

4.7.3 El campo "Content-Transfer-Encoding"

Como ya se ha indicado, los agentes SMTP y las pasarelas de correo pueden ejercer fuertes restricciones sobre los contenidos de los mensajes que se pueden transmitir fiablemente. Los tipos MIME descritos arriba son una lista de un variado conjunto de distintos tipos de objetos que se pueden incluir en un correo y la mayoría de ellos no se hallan dentro de estas restricciones. Por lo tanto, es necesario codificar estos datos de forma que se puedan transmitir y decodificar a su recepción. RFC 1521 define dos formas fiables de codificación. La razón de que haya dos formas y no una es que no es posible, dado el pequeño conjunto de caracteres fiables, desarrollar una sola forma que codifique tanto texto con un impacto mínimo en la legibilidad y datos binarios de un modo lo bastante compacto como para ser práctico.

Estas dos codificaciones se usan sólo para los cuerpos, no para las cabeceras. La codificación de cabeceras se describe en Usando caracteres no-ASCII en las cabeceras de los mensajes. El campo *"Content-Transfer-Encoding:"* define la codificación usada. Aunque engorroso, este campo enfatiza el hecho de que la codificación es una característica del transporte y no una propiedad intrínseca del objeto enviado. A pesar de haber sólo dos formas de codificación, este campo puede tomar *cinco* valores(no sensibles a mayúsculas y minúsculas). Tres de ellos especifican que no se ha realizado ninguna codificación; la diferencia entre ellos reside en que cada uno implica distintas razones por las que no ha habido codificación. Es un punto sutil pero importante. SMTP no es el único agente de correo posible de MIME, a pesar de la popularidad de sistemas de correo SMTP en Internet. Por ello, MIME permite la transmisión de datos no fiables por los estándares SMTP(STD 10/RFC 821). Si un correo de esta clase alcanza una pasarela para pasar a un sistema más restrictivo, el mecanismo de codificación especificado permite a la pasarela decidir, para cada ítem de correo, si el cuerpo se debe codificar para transmitirlo fiablemente.

Las cinco "codificaciones" son:

- "7bit"(por defecto si se omite la cabecera "Content-Transfer-Encoding:").
- "8bit"
- "Binary"
- "Quoted-Printable"
- "Base64"

Se describen en la siguiente sección.

4.7.3.1 Codificación 7bit

Codificación 7bit significa que no se ha hecho ninguna codificación y que el cuerpo consiste en líneas de texto ASCII con longitud no mayor de 1000 caracteres. Por ello, es fiable para cualquier sistema de correo que siga *estrictamente* las normas STD 10/RFC 821(por defecto, ya que son las restricciones que se aplican a los mensajes pre-MIME STD 11/RFC 822).

Nota: esta codificación *no* garantiza la total fiabilidad de los contenidos por dos razones. Primero, porque las pasarelas a redes EBCDIC tienen un conjunto de caracteres fiables más pequeño, y segundo, porque hay muchas implementaciones que no siguen SMTP.

4.7.3.2 Codificación 8bit

Codificación 8bit implica que las líneas son lo bastante cortas para el transporte SMTP, pero que puede haber caracteres no ASCII. Esta codificación sólo es posible en los agentes SMTP que soporten el *SMTP Service Extension for 8bit-MIMEtransport("Extensión SMTP para transporte MIME de 8 bits")*, descrito en el RFC 1652. En otro caso, las implementaciones de SMTP deberían poner el bit de orden superior a cero, de modo que la codificación 8bit no sería válida.

4.7.3.3 Codificación "Binary"

Indica que pueden aparecer caracteres no-ASCII y que las líneas pueden ser demasiado largas para el transporte SMTP(es decir, puede haber secuencias de 999 ó más caracteres sin una secuencia CRLF). En la actualidad no estándares para el transporte de datos binarios sin codificar en sistemas de correo de TCP/IP, por lo que el único caso en el que se puede usar codificación "binary" en un mensaje MIME en una red TCP/IP es en la cabecera de un parte externa de un cuerpo. Se podría usar en otros casos si MIME se empleara junto con otros mecanismos de transporte o con una extensión hipotética de SMTP.

4.7.3.4 Codificación "Quoted-Printable"

Es la primera de las dos codificaciones propiamente dichas y su fin es mantener la máxima legibilidad de los ficheros de texto en su forma codificada.

- Representa los caracteres no fiables con la forma hexadecimal de sus valores ASCII.
- Para mantener la longitud de cada línea a 76 caracteres o menos introduce saltos de línea reversibles ("soft").

Esta codificación utiliza el signo igual para señalar estos dos casos. Tiene cinco reglas que son:

1. Cualquier carácter excepto uno que sea parte de una secuencia de nueva línea(es decir, X'0D0A' en un fichero de texto) se puede representar con "=XX" donde XX son dos dígitos hexadecimales en mayúscula. Si ninguna de las demás reglas se aplica, el carácter debe representarse así.
2. Cualquier carácter en el rango X'21' a X'7E' excepto X'3D' ("=") se puede representar en ASCII.
3. Los caracteres ASCII TAB (X'09') y SPACE (X'20') se pueden representar en ASCII excepto cuando son el último carácter de la línea.
4. Un salto de línea se debe representar con la secuencia <CRLF>(X'0D0A'). Al codificar datos binarios, X'0D0A' no es un salto de línea y se debería codificar, según la regla

1, como "=0D=0A".
5. Las líneas codificadas no pueden tener más de 76 caracteres(sin contar el <CRLF>). Si una línea es mayor, se debe introducir un salto de línea reversible en o antes de la columna 75. Se representa con la secuencia "=<CRLF>"(X'3D0D0A').

Este esquema es un compromiso entre legibilidad, eficiencia y robustez. Como las reglas 1 y 2 utilizan la expresión "se pueden codificar", las implementaciones tienen bastante libertad a la hora de decidir a cuántos caracteres les aplican estas reglas. Si se aplican en el mínimo número de caracteres, la codificación funcionará con agentes SMTP ASCII fiables. Para pasarelas EBCDIC fiables, se añade el siguiente conjunto de caracteres ASCII:

```
! " # $ @ [ \ ] ^ ` { | } ~
```

a la lista de caracteres a los que les son aplicables las reglas. Para una robustez total, es mejor aplicar las reglas a *todos* los caracteres, excepto al conjunto de 73 caracteres que no varía al atravesar la pasarela, es decir, las letras(A-Z,a-z), los dígitos(0-9) y los siguientes 11 caracteres:

```
' ( ) + , - . / : = ?
```

Nota: Esta lista de invariantes ni siquiera incluye el carácter SPACE. Con fines prácticos, al codificar ficheros de texto sólo se debería marcar con las reglas un SPACE al final de la línea. De otra forma, la legibilidad se ve seriamente afectada.

4.7.3.5 Codificación "Base64"

Esta codificación se destina a datos que no consisten principalmente en texto. Trata el flujo de entrada como un flujo de bits, reagrupando los bits en bytes más cortos, que luego rellena hasta 8 bits para traducirlos a caracteres fiables. Como se indicó en la sección previa, sólo hay 73 caracteres fiables, por lo que la longitud máxima utilizable de cada byte es de 6 bits, que se pueden representar con sólo 64 caracteres(de hay el nombre "Base64"). Como tanto la entrada como la salida son flujos de bytes, la codificación se debe hacer en grupos de 24 bits(3 de entrada y cuatro de salida). El proceso se puede ver de la siguiente forma:

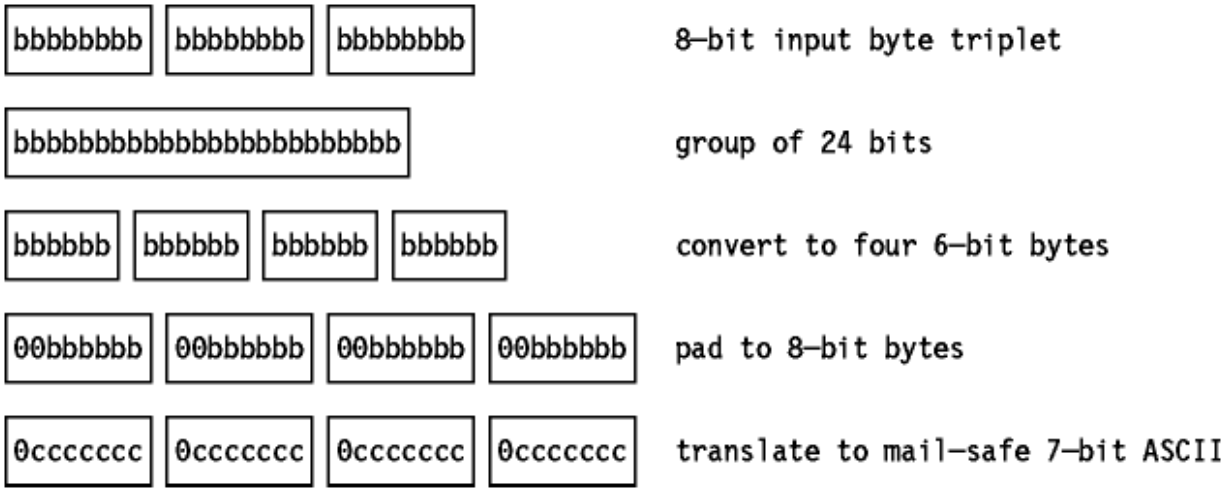


Figura: Codificación "Base64" - Conversión de 3 bytes de entrada a 4 bytes de salida en el esquema de codificación "Base64".

La tabla de traducción usada se llama el *alfabeto "Base64"*.

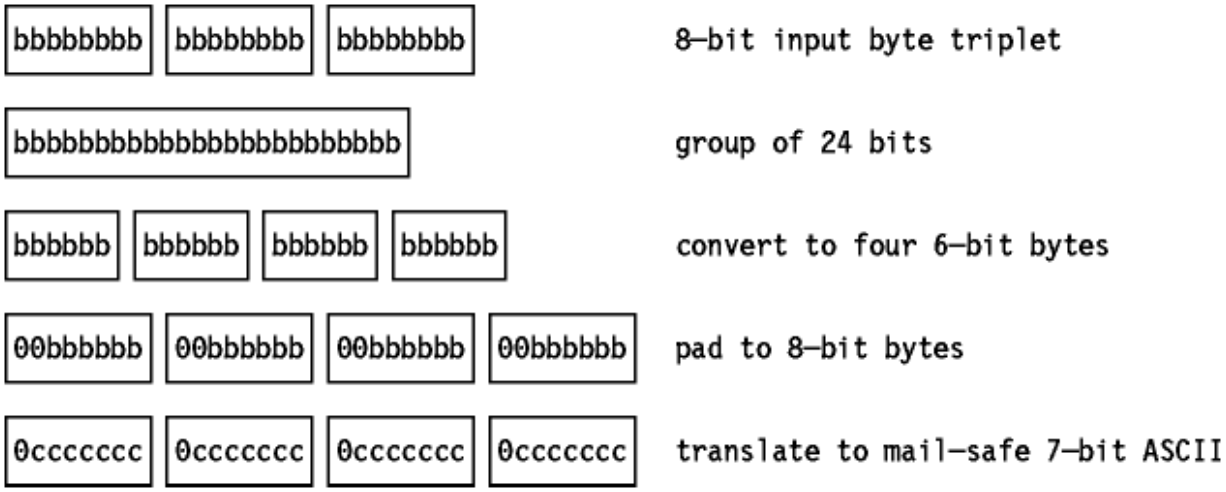


Figura: El alfabeto "Base64"

Se necesita un carácter adicional(el "=") para el relleno. Como la entrada es un flujo de bytes que se codifica en grupos de 24 bits, le podrán faltar 0, 8 ó 16 bits, al igual que a la salida. Si la salida tiene la longitud adecuada, no hace falta relleno. Si a la salida le faltan 8bits, esto se corresponde con una salida de cuarto de dos bytes completos, un "short byte" y un byte faltante. El "short byte" se rellena con los 2 bits de orden inferior a cero. Los dos bytes faltantes se sustituyen con un carácter "=". Si a la salida le faltan 16 bits, esto se corresponde con una salida de cuarto de un byte completo, un "short byte" y dos bytes faltantes. El "short byte" se rellena con los 6 bits de orden inferior a cero. Los dos bytes faltantes se sustituyen con un carácter "=". Si se utilizaron "cero caracteres", el agente receptor no sería capaz de decir al decodificar el flujo de entrada si X'00' caracteres de cola en la última o en las dos últimas posiciones eran datos orelleno. Con caracteres de relleno, el número de "="s (0, 1 o 2) da la longitud del flujo de entrada en módulo 3(0, 2 ó 1 respectivamente).

4.7.3.6 Conversión entre codificaciones

La codificación "Base64" se puede traducir libremente de y a la forma "binary" sin ambigüedad ya que ambas tratan los datos como flujos de bytes. Esto se cumple también en el caso de la traducción de "Quoted-Printable" a cualquiera de las otras dos(en el caso de la conversión a "binary", se puede ver como un proceso con una traducción con una codificación "binary" intermedia), convirtiendo las secuencias de caracteres marcados a su forma de 8 bits, borrando los saltos de línea reversibles y sustituyendo los saltos de línea por secuencias <CRLF>. Esto no es estrictamente cierto para el proceso inverso ya que la codificación "Quoted-Printable" está orientada a registros: hay una diferencia entre un salto de línea y una secuencia "=0D=0A" embebida(por ejemplo, se mapearían de distinta forma en un sistema EBCDIC).

4.7.3.7 Codificaciones múltiples

MIME *no* permite el anidamiento de codificaciones. Cualquier "Content-Type" que indique recursivamente otros campos Content-Type no puede usar un "Content-Transfer-Encoding" que no sea "7bit", "8bit" o "binary". Todas las codificaciones se deben hacer en el nivel más interno. El fin de esa restricción es simplificar el uso de los agentes de correo. Si no se permiten codificaciones anidadas, la estructura de todo el mensaje siempre le es visible para el agente sin tener que decodificar capas externas del mismo.

Esta simplificación para los agentes de correo tiene un precio: complejidad para las pasarelas. Como un agente puede especificar codificación "8bit" o "binary", una pasarela a una red en la que estas codificaciones no sean seguras tendrán que codificar el mensaje antes de pasárselo a la segunda red. La solución obvia, codificar el cuerpo del mensaje y cambiar el campo "Content-Transfer-Encoding:", no está permitida en los tipos "multipart" o "message" ya que violaría las restricciones indicadas más arriba. La pasarela, por tanto, debe descomponer el mensaje en sus componentes y recodificar las partes internar cuando sea necesaria.

Hay todavía otra restricción: en los tipos "message/partial" la codificación debe ser *siempre* "7bit"(no se permiten "8bit" y "binary"). La razón para esto es que si una pasarela necesita recodificar un mensaje, requiere disponer todo el mensaje, pero puede que no todas las partes del mismo estén disponibles(las partes se pueden transmitir en serie, debido a que la pasarela no sea capaz de almacenar todo el mensaje o incluso se pueden haber encaminado independientemente por diferentes pasarelas). Por ello, las partes de los cuerpos de los mensajes "message/partial" deben ser fiables hasta en el peor caso; es decir, deben codificarse en "7bit".

4.7.4 Usando caracteres no ASCII en cabeceras de mensajes

Todos los mecanismos estudiados arriba se refieren exclusivamente a los cuerpos, y no a las cabeceras. Los contenidos de las cabeceras de los mensajes aún se han de codificar en US - ASCII. En cabeceras que incluyan texto legible, este juego de caracteres no es apto para lenguajes distintos del inglés. Un mecanismo para incluir caracteres nacionales está definido en la segunda parte de MIME(RFC 1522). Este mecanismo difiere de la codificación "Quoted-Printable", que se usaría en el cuerpo del mensaje por las siguientes razones:

- El formato de las cabeceras esté codificado estrictamente en RFC 822, por lo que la codificación de la cabecera debe trabajar dentro de unos márgenes más estrechos que los del cuerpo.
- Los programas de retransmisión de mensajes suelen cambiar las cabeceras, por ejemplo reordenando sus campos, borrando algunos de ellos, reordenando los buzones dentro de las listas de correo o dispersando campos por distintas posiciones del mensaje original.
- Algunos programas de manipulación de mensajes no manejan correctamente algunas de las características más antiguas de RFC 822(como el uso de "\" para marcar caracteres especiales como "<" y ">.").

El enfoque de MIME es reservar secuencias improbables de caracteres ASCII legales que no son sintácticamente importantes en RFC 822. Las palabras de los campos de cabecera que tienen caracteres nacionales se reemplazan con *palabras codificadas* que tienen la forma:

```
=?charset?encoding?word?=
```

donde:

charset
: es el valor admitido para el parámetro "charset" usado con el tipo MIME "text/plain", es decir, "us-ascii", "iso-8859-1" a "iso-8859-9".

encoding
: "B" o "Q. "B" es idéntica a la codificación "Base64" aplicada a los cuerpos. "Q" es similar a "Quoted-Printable" pero utiliza "_" para representar X'20'(SPACE). La codificación (18) Q requiere codificar los caracteres "_" y no permite saltos de línea. Cualquier carácter ASCII diferente de "_", "=" y SPACE no tienen que marcarse en la palabra codificada a menos que sea sintácticamente significativo cuando cabecera se descompone por RFC 822. *"charset"* y *"encoding"* no son sensibles a mayúsculas y minúsculas.

word
: es una ristra de ASCII caracteres ASCII distintos de SPACE de acuerdo a las reglas de la codificación dada.

Una palabra codificada no debe tener caracteres en blanco embebidos(SPACE o TAB), puede tener hasta 75 caracteres, y puede no estar en una línea de más de 76 caracteres(sin contar <CRLF>). Estas reglas aseguran que las pasarelas no plegarán las palabras codificadas por el medio. Las palabras codificadas, en general, se pueden usar en las partes legibles de los campos de la cabecera. Por ejemplo, si un buzón se especifica de la forma:

```
The Octopus <octopus@garden.under.the.sea>
```

se podría utilizar una palabra codificada en la sección "The Octopus" pero no en la parte de dirección entre el "<" y el ">"). RFC 1522 especifica precisamente donde se pueden usar palabras codificadas en relación con la sintaxis RFC 822.

4.7.5 Referencias

Se puede encontrar una descripción detallada de MIME en los siguientes RFCs:

- *RFC 1521 -- MIME ("Multipurpose Internet Mail Extensions") Parte uno: Mecanismos para especificar y describir el formato del de los mensajes de Internet.*
- *RFC 1522 -- MIME ("Multipurpose Internet Mail Extensions") Parte dos: Extensiones de la cabecera de los mensajes para texto no ASCII*

Tabla de contenidos REXEC ("Remote Execution Command Protocol")

4.8 REXEC("Remote Execution Command Protocol")

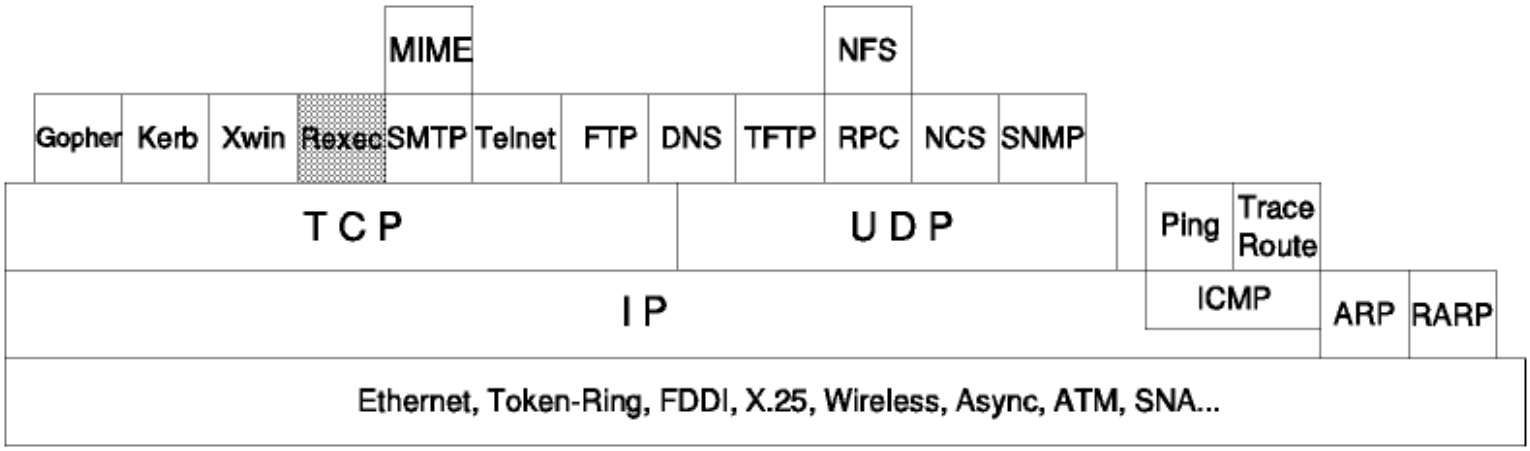


Figura: REXEC("Remote Execution Command Protocol")

REXECD (*Remote EXEcution Command Daemon*) es un servidor que permite la ejecución del comando REXEC o RSH ("Remote Shell Protocol") desde un host remoto sobre la red TCP/IP. La función de cliente es realizada por el proceso REXEC.

4.8.1 Principio de funcionamiento

REXECD es un servidor(de tipo "daemon" o programa residente). Maneja comandos lanzados por host remotos, y transfiere órdenes a máquinas virtuales que actúan como esclavos para la ejecución de tareas. El "daemon" realiza un login automático, así como la autentificación cuando el usuario introduce su identificador y password.

El comando REXEC se usa para definir el identificador de usuario, el password, dirección del host, y el proceso a iniciar en el host remoto. Tanto el servidor como el cliente están conectado sobre la red TCP/IP.

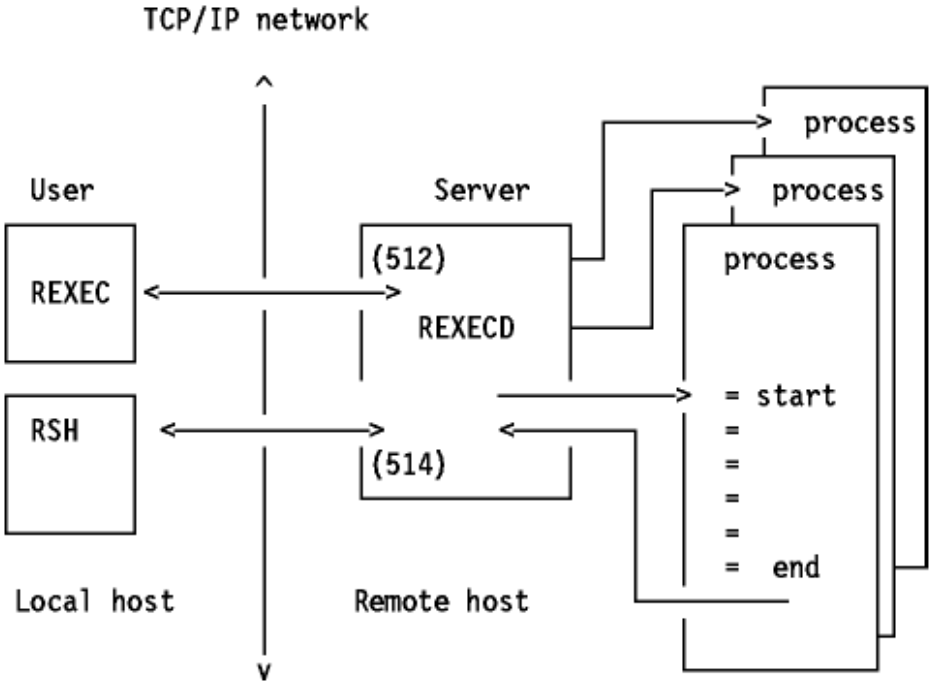


Figura: Principio de REXECD

4.9 El sistema X Window

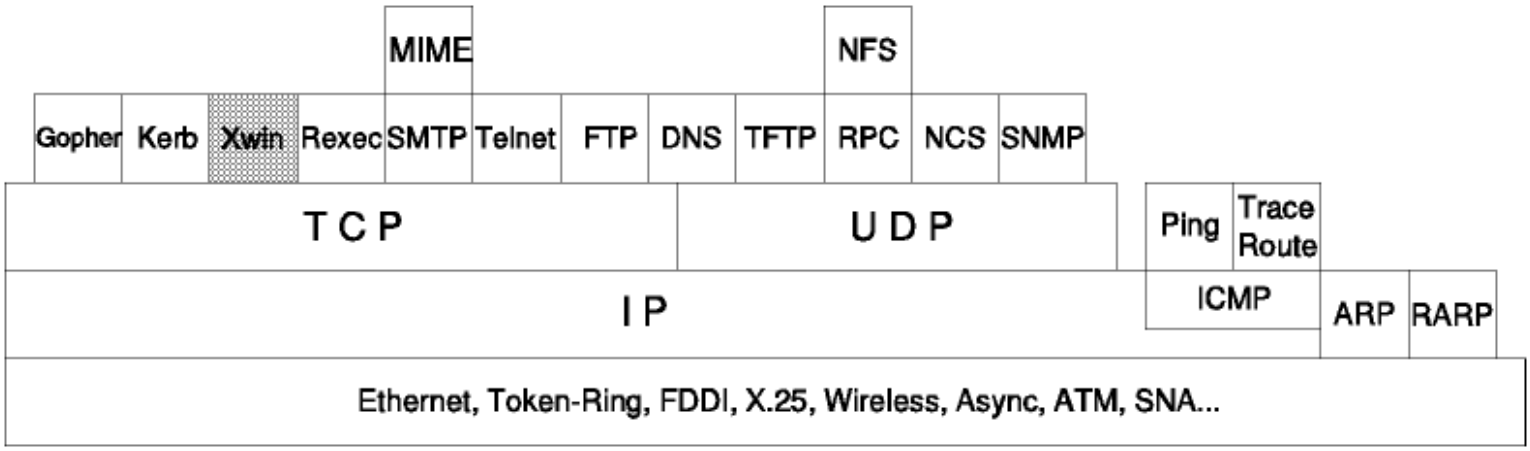


Figura: Sistema X Window

El sistema X Window(referido de ahora en adelante como X) es una de las *GUI("Graphical User Interface")*, o sistema de ventanas de mapas de bits, más usadas. Todos los principales distribuidores de estaciones de trabajo trabajan con ella, y es usada cada vez más por los usuarios de todo el mundo. X ofrece es más que un simple entorno. Ofrece además una plataforma para paquetes comerciales incorporados en ella de modo exclusivo. Algunos grupos industriales, además de producir software de aplicación, han creado paquetes de software propios y estándares de interfaces que refinan las capacidades el sistema X Window. Estos paquetes se integran en las aplicaciones para mejorar su *look and feel*. Los dos paquetes comerciales más significativos en esta área son el *MOTIF* de OSF("Open Software Foundation") y el *Open Look* de UNIX International. X fue una creación de Robert Scheifler, Jim Gettys, y otros en el MIT, como parte del *Proyecto Atenea*, un proyecto de investigación dedicado al chequeo de grande redes de ordenadores personales y estaciones de trabajo. (Para más información sobre este proyecto, remitirse a *Proyecto Atenea: estudio de computación distribuída en el MIT*). Como parte de este estudio, se consideró necesario un sistema de entorno de ventanas que unificase los entornos existentes. X se concibió como tal, de modo que fuera un sistema de ventanas que pudiera ser usado en las diferentes redes y ordenadores integrados en el proyecto.

A medida que el proyecto Atenea progresaba, X evolucionó hacia un sistema de ventanas transportable y basado en red. Gran parte del trabajo inicial desarrollado en X se derivó de un sistema de ventanas de la misma época, desarrollado por Stanford y llamado W. De hecho, el nombre X fue simplemente una continuación del nombre anterior, W. El Consorcio del MIT, fundado en 1988, se dedica al desarrollo de X y a promover la cooperación en la industria informática para estandarizar las interfaces de X.

Las versiones actuales de X contienen dos números: el *número de versión* que indica el protocolo principal o revisiones del estándar, y el *número de versión*, que indica cambios menores. En el momento de redactar este documento, la última versión es X11R6; V1.2 para el OSF/MOTIF(basado en X11R5). Las principales revisiones de X son incompatibles entre sí, aunque hay compatibilidad hacia atrás con versiones menores dentro de una misma categoría.

El objeto de X fue permitir al usuario controlar todas las sesiones desde una sola pantalla, con aplicaciones ejecutándose bien en una ventana, o en terminales virtuales separadas, pero con un icono en la ventana principal recordándole la existencia de esa aplicación.

X proporciona la capacidad de gestionar tanto ventanas locales como remotas. Las ventanas remotas se establecen a través de TCP/IP, y las locales mediante el uso de *zócalos* de BSD.

4.9.1 Concepto funcional

Básicamente hay dos partes que se comunican entre sí:

1. La aplicación, que recibe entradas del usuario, ejecuta código y envía la salida al usuario. En vez de leer y escribir directamente en el display, la aplicación usa la interfaz de programación Xlib para enviar y recibir datos de/a la terminal de usuario. La parte de aplicación se conoce también como cliente X.
2. La terminal de usuario, que ejecuta la interfaz con el display y envía o recibe datos a/de la aplicación se denomina servidor X.

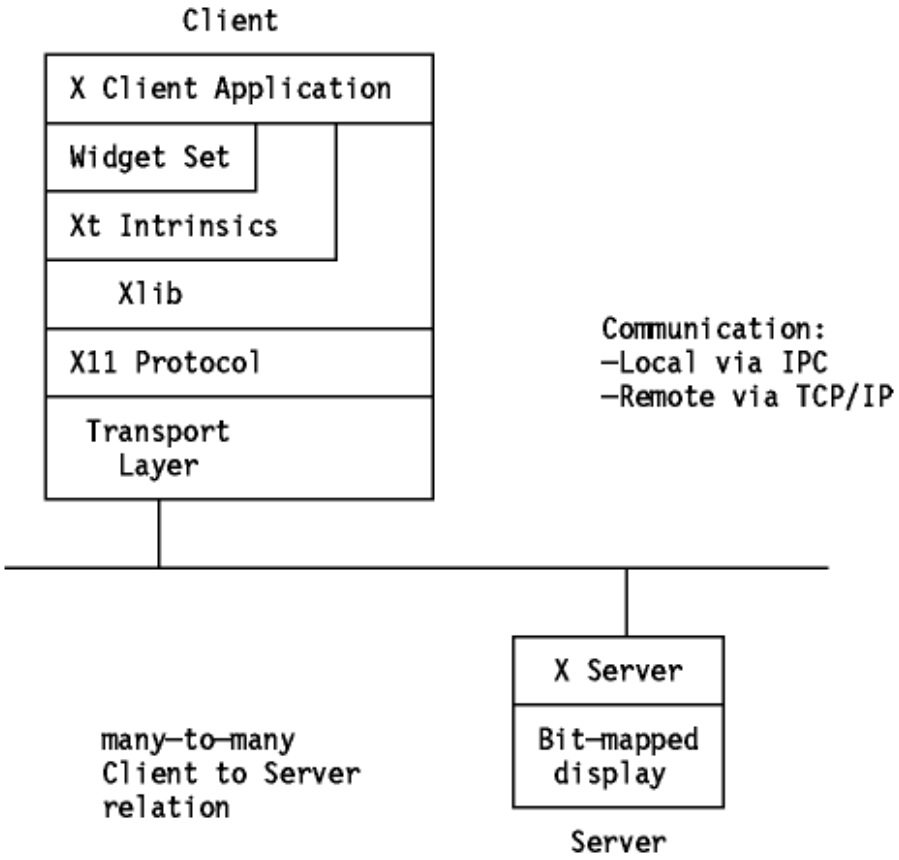


Figura: Concepto del sistema X Window - Clientes y servidores comunicándose.

Terminología:

- **Servidor X**: Es un programa dedicado a suministrar servicios de display en una terminal gráfica, en favor del usuario y a petición del cliente X del usuario. Controla la pantalla y maneja el teclado y el ratón(u otros dispositivos de entrada) para uno o más clientes X. Igualmente, es responsable de la salida sobre el display, el mapeado de colores, la carga de fuentes y el mapeado del teclado. Típicamente, los servidores X se ejecutan en PCs y terminales de trabajo de tipo gráfico y de alto rendimiento, además de en las "terminales X", diseñadas para ejecutar sólo servidores X.

  El servidor X X11R5 ha proporcionado algunas mejoras en la velocidad y nuevas interfaces para la fuentes. Existe un *Protocolo de Servicio de Fuentes X* disponible para permitir a los servidores X delegar la gestión de las fuentes en un servidor de fuentes.

- **Cliente X**: Es la *aplicación* propiamente dicha y está diseñado para emplear una interfaz gráfica de usuario para mostrar sus salidas. Típicamente, muchos clientes X compiten por los servicios de un servidor X por cada usuario y display. El gestor de X Window, que es una entidad separada y diferenciada, resuelve los conflictos producidos por esta competencia. *Xterm* y *Xclock* son dos ejemplos de clientes X.

  X11R5 ha añadido algunos clientes más, nuevas demos y una implementación completamente nueva del mapa de bits y xmag.

- **El gestor de X Window**: Es un cliente X localizado en la estación de trabajo donde se ejecuta el servidor X. El gestor no es necesario para crear las ventanas, aunque permite redimensionarlas, moverlas y modificarlas cuando sea solicitado.
- **Protocolo X**: Se ejecuta sobre la conexión de red y permite que se efectúen solicitudes y respuestas entre cliente y servidor. Está orientado a conexión(usa TCP) y describe el formato de los mensajes intercambiados entre cliente y servidor sobre la conexión.
- **Xlib**: Contiene una rudimentaria interfaz de programación de aplicación. Se trata de una colección de subrutinas primitivas de C embebidas en todos los clientes X, que proporciona el acceso de más bajo nivel al protocolo X. Los procedimientos en Xlib traducen las peticiones de los clientes a solicitudes del protocolo X y analizan los mensajes que llegan(eventos, respuestas, errores) de los servidores X, además de suministrar diversas utilidades adicionales, por ejemplo, operaciones independientes del sistema operativo, como puede ser la gestión de almacenamiento. Es posible escribir aplicaciones enteras con Xlib. De hecho, la mayoría de los clientes X existentes son o han sido desarrollados de este modo.

  X11R5 ha añadido dos grandes funcionalidades a Xlib:

  - Dispositivos independientes del color
  - Internacionalización (i18n): significa que los clientes X pueden adaptarse a los requerimientos de distintos lenguajes nativos, costumbres locales y códigos de caracteres.
- **X Toolkits**: La complejidad de la interfaz de bajo nivel Xlib y de protocolo X subyacente puede ser manejada con un creciente variedad de *Toolkits* disponibles. Los X Toolkits son librerías de software que añden funcionalidades de alto nivel para implementar *objetos* comunes de la interfaz de usuario, como por ejemplo botones, menúes, barras de desplazamiento, así como herramientas de configuración para organizar estos objetos en el display. El estándar de X del MIT proporciona la base para el desarrollo de los X Toolkits. La librería, llamada **Intrínsecas de Xlib o Xt,** constituye las piezas elementales para construir una serie de objetos de la interfaz llamados *widgets*.
- Widgets: Para toolkits basados en Xt, se hace uso de mecanismos de interfaz comunes, llamados *widgets*. Un *widget set* es básicamente una ventana X más algunos datos adicionales y un conjunto de procedimientos para operar sobre ellos. Los widgets son un concepto exclusivo del cliente. Ni el servidor X ni el protocolo son capaces de entender los widgets. Un ejemplo de widget set es el *Xaw*, más conocido como *Athena Widget Set*, distribuído por el MIT con el código fuente de X11.

Funcionalidad:

- Los servidores y clientes X pueden estar en hosts diferentes. En ese caso podrán usar TCP/IP para comunicarse en la red. También pueden estar en la misma máquina, usando IPC(comunicación entre procesos) para comunicarse(a través de zócalos).
- Sólo hay un servidor X por terminal, con el que se pueden comunicar múltiples clientes X. El deber del servidor X es mostrar las ventanas de aplicación y enviar al cliente X correspondiente las entradas del usuario.
- Depende del cliente X el mantener las ventanas que ha creado. Los cambios efectuados en el display por otros clientes son notificados por *eventos* del servidor X. Sin embargo, los clientes no tienen que preocuparse de qué parte sus ventanas son visibles y cuáles no, o de cuándo se deben dibujar o redibujar sus ventanas.

- El servidor X lleva la cuenta de las ventanas visibles y no visibles manteniendo *pilas o stacks*. Una pial contiene los *"hijos de primera generación"* de una ventana padre. Una ventana hijo puede ser padre padre si tiene a su vez una o más ventanas hijo, que de nuevo se almacenarán en una *subpila*. La *pila primaria* es la que guarda todas las ventanas localizadas directamente bajo la raíz. Ver Figura - Estructura del sistema X Window para una ilustración. Las subventas sólo pueden ser visibles del todo cuando su padre está en la cima de su respectiva pila y se halla mapeada en el display.
- El servidor X en sí carece de funciones de gestión; sólo realiza el recorte de las ventanas sobre el puerto de visión según sus pilas. Cada cliente es responsable de sus propias ventanas. Hay un *gestor de ventanas("Window Manager")* que manipula las ventanas de la cima de todos los clientes. El gestor no forma parte del servidor X sino que es en sí un cliente. En cuanto el gestor cambia algo en la pantalla(por ejemplo, redimensionando una ventana), hace que el servidor X envíe un evento a los demás clientes.
- Los clientes envían *mensajes de solicitud* al servidor, que contesta con *mensajes de respuesta* o *mensajes de error*. El servidor X también puede enviar mensajes de eventos a las aplicaciones. Los mensajes de eventos indican cambios en las ventanas(y su visibilidad), y en la entrada de usuario(ratón y teclado).

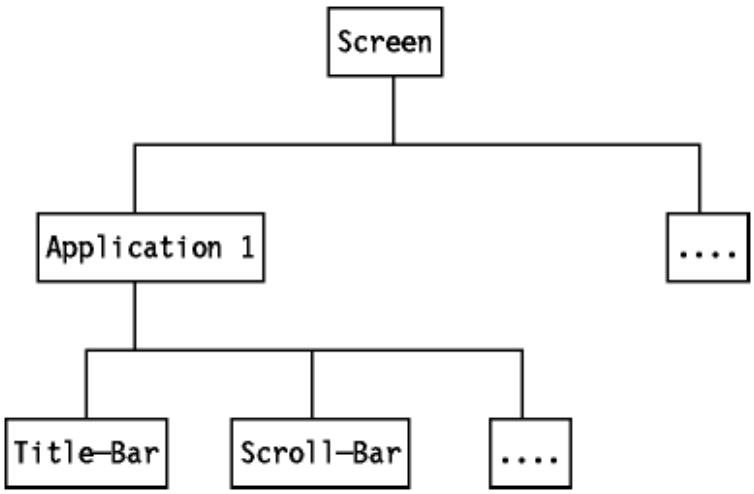


Figura: Estructura de ventanas del sistema X Window - Cada ventana es hijo de otra ventana, siendo la raíz el display.

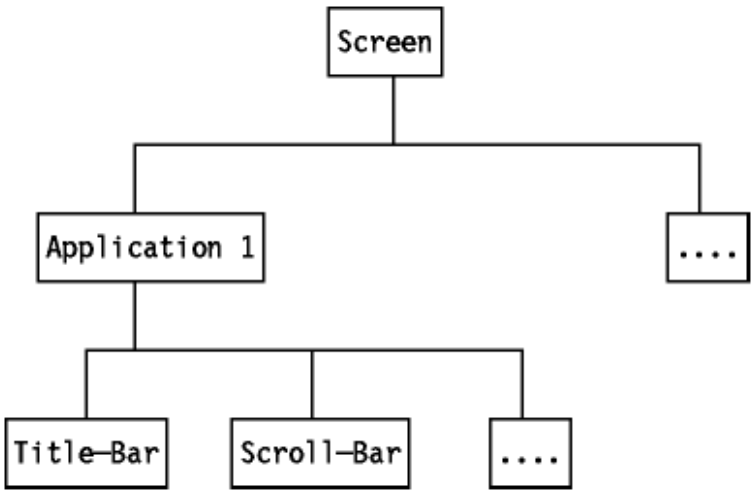
Aplicar el concepto de cliente/servidor tiene las siguientes ventajas:

- Las aplicaciones no tienen que conocer las características hardware de la terminal.
- Las aplicaciones no tienen que estar en el mismo ordenador que la terminal.
- Los programas escritos en Xlib son portables.
- Se pueden añadir nuevos tipos de terminal sólo con proporcionar un servidor X adecuado.
- Los porgramadores no tienen que ocuparse de las comunicaciones, sólo han de escribir aplicaciones gráficas para Xlib, con independencia de si los usuarios son locales o remotos.

4.9.2 Protocolo

Un protocolo X se puede implementar sobre cualquier mecanismo de transporte fiable y orientado a conexión. Utiliza un simple bloque de protocolo sobre la capa de transporte. Se usan cuatro tipos de mensajes:

- Formato de petición. Solicita el flujo del cliente al servidor.



Donde:

  - "Major" y "minor" son códigos de operación de 1 byte cada uno.
  - "Length" vale 2 bytes.
  - "Data" puede tener cero o más bytes de longitud, según la petición.
- Formato de respuesta : bloque de 32 bytes.
- Formato de error: bloque de 32 bytes.
- Formato de evento: bloque de 32 bytes.

Los mensajes de respuesta, error y evento son enviados por el servidor al cliente.

Los displays se numeran siempre a partir de cero. Para conexiones TCP, el número de display N se asocia con el puerto 5800+N (hex 5800) y el 5900+N. El servidor X trata las conexiones en el puerto 58xx como conexiones con hosts que usan el formato "primero el byte inferior", y a los 59xx como aquellos que tienen el formato "primero el byte de orden superior".

Hay más de cien posibles peticiones, cada una correspondiente a una llamada de aplicación de *Xlib*. Ya que este documento no es una guía del programador, no trataremos las funciones de *Xlib*. El *RFC 1013 - Protocolo del sistema X Window, Version 11* contiene la actualización alfa de 1987 del X11. Para conseguir documentación sobre la versión actual de X11R6, ponerse en contacto con el MIT o con un distribuidor de libros técnicos.

 Tabla de contenidos  RPC ("Remote Procedure Call")

4.10 RPC("Remote Procure Call")

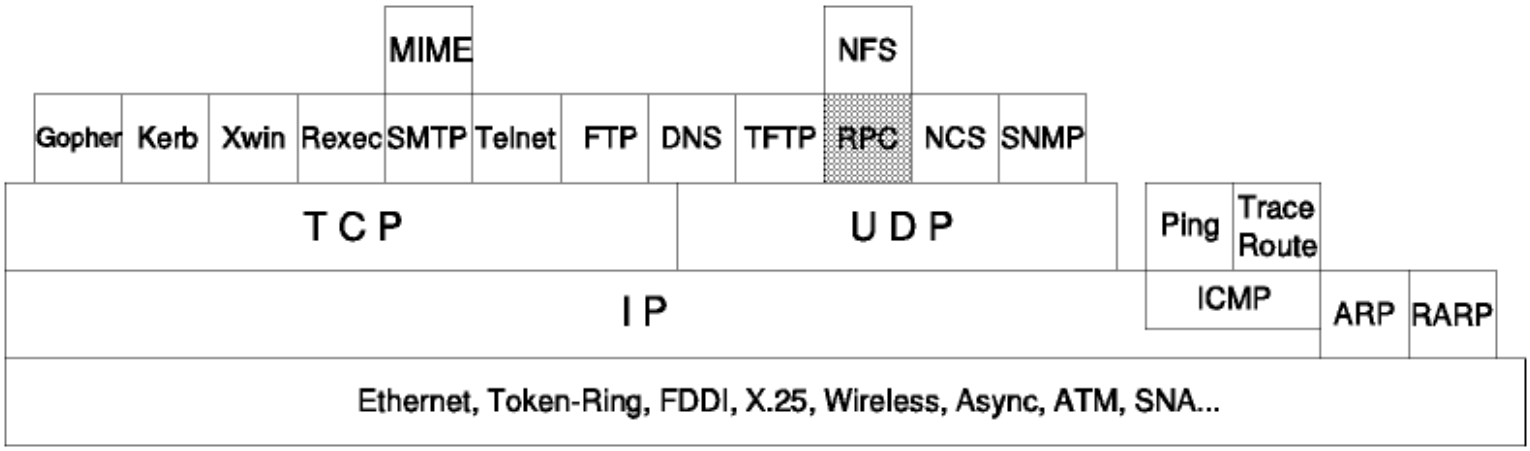


Figure: RPC("Remote Procure Call")

El RPC de Sun es un *protocolo propuesto como estándar*. Su status es *electivo*.

RCP es un estándar desarrollado por Sun Microsystems y usado por muchos distribuidores de sistemas UNIX. La especificación actual de UNIX se halla en el *RFC 1057 - RPC ("Remote Procure Call"): especificación de protocolo de la versión 2*.

El RPC es una interfaz de programación de aplicación(API) disponible para el desarrollo de aplicaciones distribuidas. Permite que los programas llamen a subrutinas que se ejecutan en un sistema remoto. El programa llamador, denominado (llamado *client*) envía una *mensaje de llamada* al proceso *proceso servidor* y espera por un *mensaje de respuesta*. La llamada incluye los parámetros del procedimiento y la respuesta los resultados.

El RPC de Sun consta de las siguientes partes:

- *RPCGEN*: Un compilador que toma la definición de la interfaz de un procedimiento remoto, y genera los "stubs" del cliente y del servidor.
- *XDR ("eXternal Data Representation")*: Una forma estándar de codificar datos de modo para que sean transportables entre distintos sistemas. Impone una ordenación big - endian de los bytes y el tamaño mínimo de cualquier campo ha de ser 32 bits. Esto significa que tanto el cliente como el servidor han de realizar algún tipo de traducción.
- Una librería "run-time".

4.10.1 Concepto de RPC

El concepto de RPC se puede simplificar del modo siguiente:

- El proceso llamador envía un mensaje de llamada y espera por la respuesta.
- En el lado del servidor un proceso permanece dormido a la espera de mensajes de llamada. Cuando llega una llamada, el proceso servidor extrae los parámetros del procedimiento, calcula los resultados y los devuelve en un mensaje de respuesta.

Ver Figura - RPC muestra un modelo concepual de RPC.

Este es sólo un posible modelo, ya que el protocolo RPC de Sun no impone restricciones específicas en el modelo de concurrencia. En el modelo anterior, el proceso llamador se bloquea hasta que se recibe un mensaje de respuesta. Otros modelos son igualmente posibles; por ejemplo, el llamador puede continuar su ejecución mientras espera una respuesta, o el servidor puede despachar una tarea separada para cada llamada que reciba de modo que quede libre para recibir otros mensajes.

Las llamadas a procedimientos remotos difieren de las llamadas a procedimientos locales en los siguientes aspectos:

- Uso de variables globales ya que el servidor no tiene acceso al espacio de memoria del llamador.
- El rendimiento puede verse afectado por los tiempos de transmisión.
- Puede ser necesaria la autentificación del usuario.
- Se debe conocer la dirección del servidor.

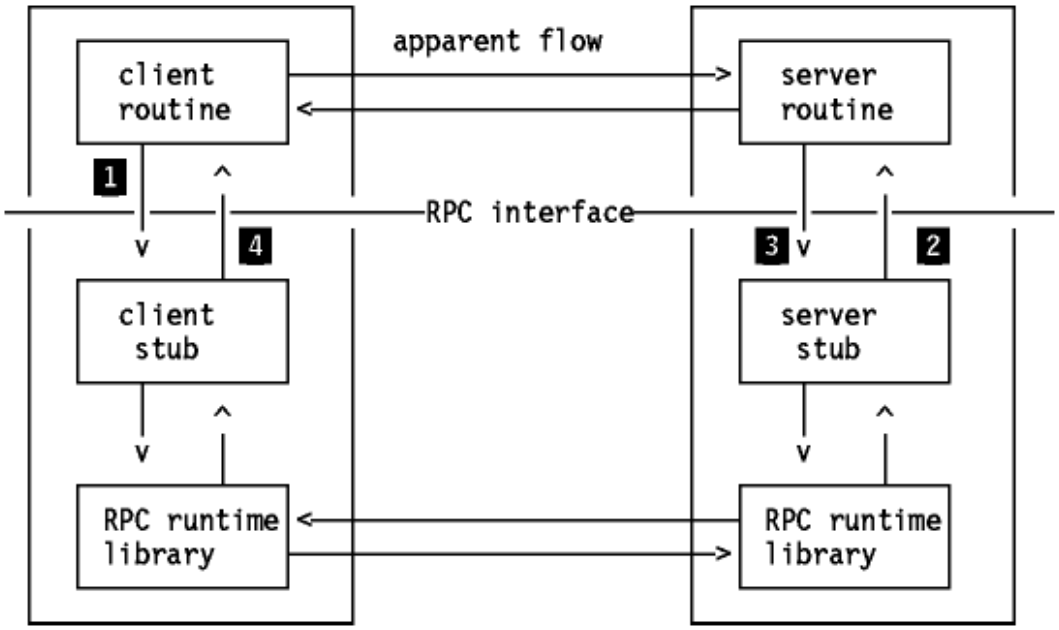


Figure: RPC - Modelo de llamadas a procedimientos remotos

4.10.1.1 Transporte

El protocolo RPC se puede implementar sobre cualquier protocolo de transporte. En el caso de TCP/IP, puede usar tanto TCP como UDP como capa de transporte. El tipo de *transporte* es un parámetro del comando RPCGEN. En caso de que se use UDP, recuérdese que no proporciona fiabilidad, por lo que dependerá del programa llamador el garantizarla(usando tiempos límite y retransmisiones, implementadas normalmente en rutinas de librería e RPC). Cabe señalar que incluso con TCP, el programa llamador sigue necesitando una rutina para el tiempo límite con el fin de tratar situaciones excepcionales, como por ejemplo la caída del servidor.

Los mensajes de llamada y respuesta se formatean al estándar XDR.

4.10.1.2 Mensaje de llamada de RPC

El mensaje de llamada de RPC consta de varios campos:

- Números de programa y de procedimiento

  Cada llamada contiene tres campos(enteros sin signo):

  - Número del programa remoto
  - Número de versión del programa remoto
  - Número del procedimiento remoto

  que identifican unívocamente al procedimiento e ejecutar. El número de programa remoto identifica un grupo funcional de procedimientos, por ejemplo, un sistema de ficheros, que incluiría procedimientos individuales como "leer" y "escribir". Estos procedimientos individuales se identifican con un número de procedimiento único dentro del programa remoto. A medida que el programa remoto evoluciona, a cada versión se le asigna un número de versión.

  Cada programa remoto está conectado a un puerto de. El número de este puerto se puede elegir libremente, exceptuando los puertos reservados para "servicios bien conocidos". Es evidente que el llamador tendrá que conocer el número de puerto usado por el programa remoto.

  Números de programas asignados:

  00000000 - 1FFFFFFF
  definidos por Sun
  20000000 - 3FFFFFFF
  definidos por el usuario
  40000000 - 5FFFFFFF
  de transición(números temporales)
  60000000 - FFFFFFFF
  reservados

- Campos de autentificación

  Existen dos campos, *credenciales* y *verificador*, para la autentificación del llamador al servicio. Depende del servidor el usar esta información para la autentificación del usuario. Además, cada implementación es libre de elegir entre los varios protocolo de autentificación que están soportados. Algunos de ellos son:

  - Autentificación nula.
  - Autentificación UNIX. Los llamadores de un procedimiento remoto se pueden identificar de igual modo que en el sistema UNIX.
  - Autentificación DES. Además del identificador de usuario, al servidor se le envía un campo correspondiente a un sello de tiempo. Este sello de tiempo es la hora actual, cifrada con una llave conocida sólo para el servidor y el llamador (basado en el concepto de *llave secreta* y *llave pública* de DES).

- Parámetros de los procedimientos.

  Los datos(parámetros) pasados al procedimiento remoto.

4.10.1.3 Mensaje de respuesta de RPC

Existen diversas respuestas, dependiendo del tipo de acción a tomar:

- SUCCESS: los resultados del procedimiento se le devuelven al cliente.
- RPC_MISMATCH: el servidor está ejecutando una versión de RPC distinta de la del llamador.
- AUTH_ERROR: autentificación de usuario fallida.
- PROG_MISMATCH: el programa no está disponible, la versión solicitada no existe o el procedimiento no está disponible.

Para un descripción detallada de los mensajes de llamada y respuesta, ver el *RFC 1057 - RPC: ("Remote Procedure Call"): especificación de protocolo de la versión 2*, que contiene además las definiciones de tipos (typedef) para los mensajes en el lenguaje XDR.

4.10.1.4 Portmap o Portmapper(Mapeador de puertos)

Como se indica más arriba, el llamador tiene que conocer el número de puerto exacto usado por un programa RCP concreto para ser capaz de enviarle un mensaje. Portmap es una aplicación del servidor que mapea el número de programa y de versión al puerto usado por ese programa. Debido a que portmap tiene asignado el número de puerto reservado (*servicio bien conocido*) 111, todo lo que tiene que hacer el llamador es preguntarle al servicio Portmap en el host remoto por el puerto usado por el programa servidor. Ver Figura - Portmap.

Portmap sólo tiene conocimiento de los programas de su host(sólo programas RPC en el host local).

Con el fin de que RPC adquiera conocimiento del RPC, cada programa RCP debería registrarse con el Portmap local cuando este arranca. También debería anular su registro cuando el Portmap finaliza su ejecución.

Normalmente, la aplicación llamadora contacta con el Portmap en el host de destino para obtener el número de puerto correcto de un programa remoto determinado, y luego envía el mensaje de llamada a ese puerto. Existe una variante consistente en que el llamador envía también los parámetros del procedimiento al Portmap y este a su vez se encarga de invocarlo.

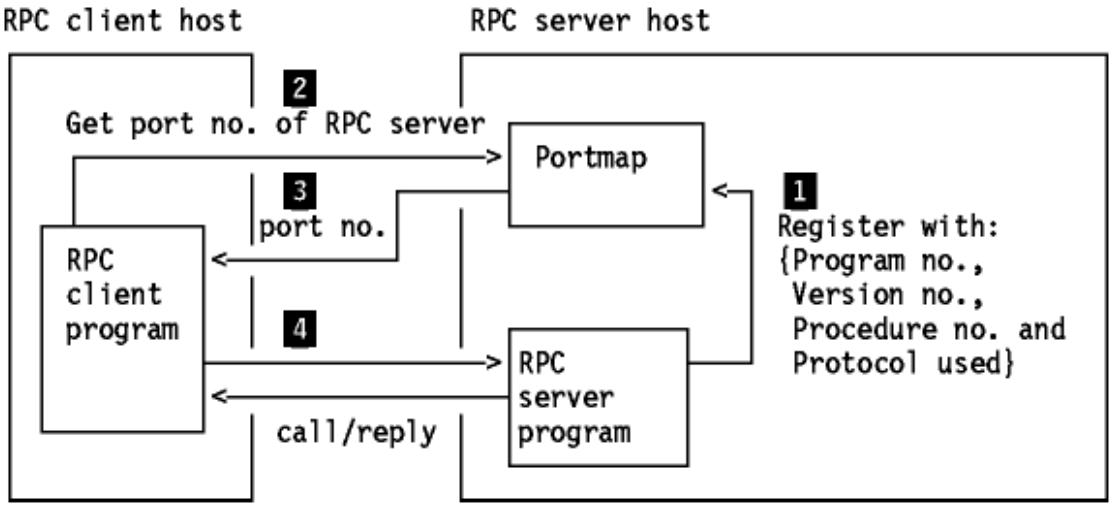


Figura: Portmap - Informa al llamador de que número de puerto ocupa un programa en su host.

4.10.1.5 RPCGEN

RPCGEN es una herramienta que genera código en C para el protocolo RPC. La entrada de RPCGEN es un fichero escrito en un lenguaje similar a C, conocido como lenguaje RPC. Asumiendo que se usa un fichero de entrada llamado *proto.x* , RPCGEN produce los siguientes ficheros de salida:

- Un fichero cabecera llamado *proto.h* que contiene definiciones comunes de constantes y macros.
- El código fuente del "stub" del cliente, *protoc.c*
- El código fuente del "stub" del servidor, *protos.c*
- El fichero fuente de rutinas XDR, *protox.c*

Tabla de contenidos NCS ("Network Computing System")

4.11 NCS("Network Computing System")

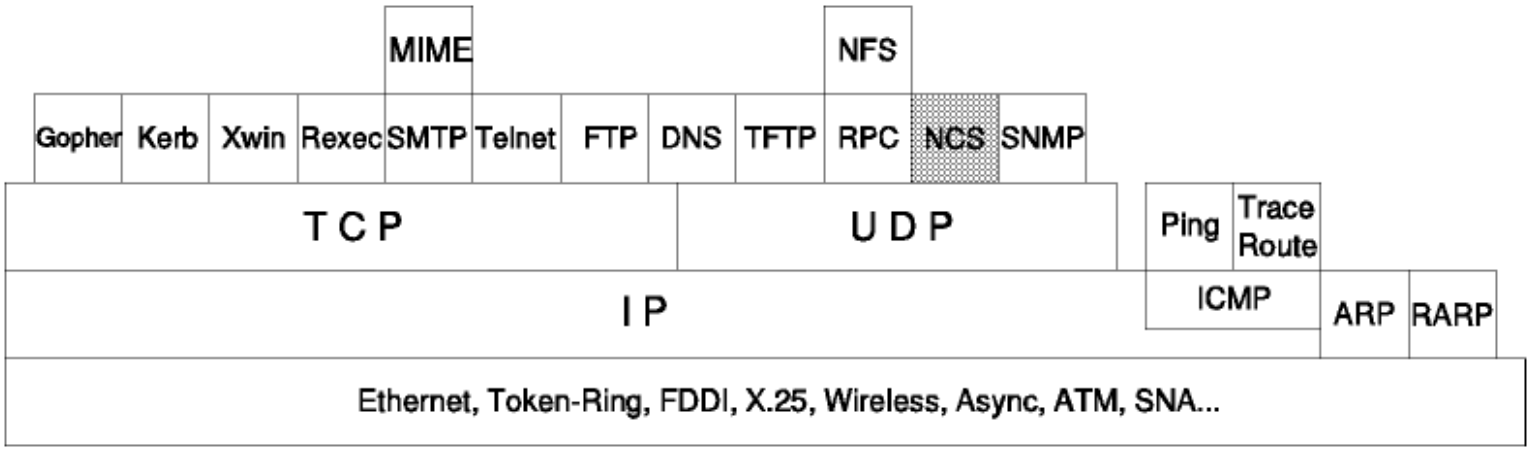


Figura: NCS("Network Computing System")

El NCS de APOLLO es una implementación de la NCA ("Network Computing Architecture") desarrollada con el fin de proporcionar herramientas para diseñar, implementar y dar soporte para aplicaciones que requieran datos y computación distribuidos. Esto se consigue mediante implementaciones de NCS sobre la interfaz de RPC, que es diferente del RPC de Sun.

El NCS está orientado a objetos. Esto permite a los programas acceder a los objetos a través de interfaces con independientemente de las máquinas con las que se comunican. Estos tipos de programas tienen un diseño más simple y son menos susceptibles frente a cambios en el hardware o en la red.

Un objeto es una entidad gestionada por operaciones definidas que tienen un tipo especificador de la clase. Por ejemplo, un fichero de disco es un objeto y puede ser de tipo ASCII.

Una interfaz es un conjunto de operaciones que manipulan los objetos.

El NCA emplea un concepto expandido llamado objetos replicados que son copias de un objeto que tienen el mismo identificador. Puede ser débilmente consistente o fuertemente consistente. Los objetos replicados débilmente consistentes pueden ser accedidos aunque no sean idénticos. Los objetos replicados fuertemente consistentes sólo pueden ser accedidos cuando son idénticos. El uso de un tipo u otro de objetos depende de los requerimientos de rendimiento, disponibilidad y consistencia.

Los datos y el procesamiento distribuidos se obtienen gracias al uso de los siguientes componentes:

1. La librería "runtime" de *RPC("Remote Procedure Call")*
2. El compilador de *NIDL("Network Interface Definition Language")*
3. El *Location Broker*

El *NCK("Network Computing Kernel")* consiste en el Location Broker y la librería "runtime" de RPC, que suministra apoyo en tiempo de ejecución a la computación distribuida. Este kernel(NCK) y el compilador son la base para el desarrollo e implementación de aplicaciones distribuidas.

4.11.0.1 NCS RPC

El NCS RPC puede utilizar los protocolos de comunicaciones de dominios de red("Domain network communications protocols(S)") y los protocolos de Internet de DARPA(UDP/IP). La selección se hace en función de la dirección de destino de tal forma que un programa puede acceder o no a un dominio.

El NCS RPC maneja el concepto de zócalo de Berkeley. Puede atender a más de un zócalo identificado por un dirección de zócalo dividida en familia de direcciones (que define la estructura de la dirección), dirección de red(dirección del host) y número de puerto(dirección de destino final).

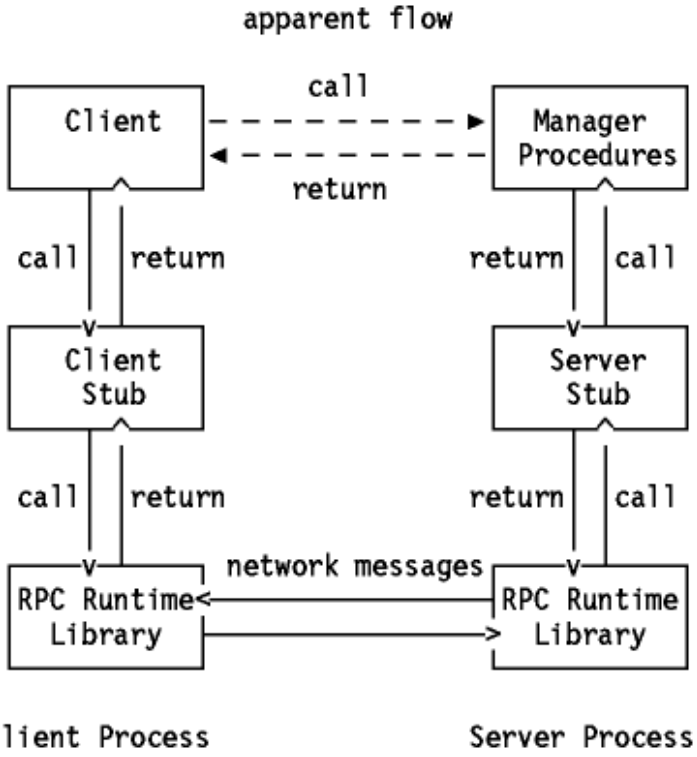


Figura: componentes de NCS

El procedimiento cliente usa convenios estándar de llamada a procedimientos, pero es ejecutado remotamente por el servidor. El programa que realiza llamadas remotas a procedimientos es el cliente RPC. No tiene idea de cómo se implementa la interfaz y puede que no conozca la dirección del servidor.

El proceso que recibe el paquete de solicitud de la operación de la librería "runtime" de RPC es el servidor RPC. Es responsable de enviar la respuesta con los resultados de la operación. Un servidor puede exportar una interfaz para más de un objeto.

El proceso cliente tiene tres componentes: el procedimiento cliente que hace llamadas, el "stub" del cliente y la librería "runtime" de RPC. El "stub" del cliente es el encargado de usar la librería "runtime" de RPC para hacer que se ejecuten las llamadas del cliente.

El proceso servidor tiene tres componentes: los procedimientos de gestión correspondientes a la aplicación cliente, el "stub" del servidor correspondiente al del cliente y la librería "runtime" de RPC. El servidor puede denominarse gestor o manager.

Cuando el cliente solicita una operación en un objeto concreto a través de RPC, debe indicar el objeto sobre el que se va a efectuar, así como el servidor que exporta la interfaz que contiene esa operación. Esta información se pasa mediante un manejador o "handle", creado y gestionado por varias llamadas que proporciona NCS. La representación del servidor en el manejador se denomina liga o enlace("binding"). El cliente puede crear o no una liga en el manejador solicitando un RPC con los siguientes estados:

- Unbound(desligado) - sin identificación(RPC hace un broadcast a todos los host en la red local y acepta la primera respuesta).
- Bound-to-host(ligado al host) - el manejador contiene identificación del host pero sin especificar un servidor(se envía un RPC al host y el Location Broker local encuentra el puerto correcto).
- Bound-to-server(ligado al servidor) - el manejador contiene la identificación completa(el RPC se envía al puerto específico el servidor).

Los "stubs" son responsables de que la llamada remota sea lo más transparente posible. Hacen de intermediarios entre el cliente y los procedimientos de gestión, convirtiendo datos para el uso de las rutinas de la librería "runtime" de RPC.

Esta librería transmite paquetes RPC que contienen rutinas, tablas y datos para apoyar la comunicación entre el cliente y el "stub" del servidor. Hay tres tipos de llamadas:

- Llamadas del cliente usadas para operar con los manejadores y enviar paquetes.
- Llamadas del servidor usadas para crear zócalos, registrar interfaces y devolver identificaciones de objetos.
- Llamadas de conversión usadas para determinar la dirección del zócalo de un host determinado y devolver el nombre del host y número de puerto para ese zócalo.

4.11.0.2 NIDL("Network Interface Definition Language")

El NIDL es un lenguaje de desarrollo que define por completo la interfaz y los parámetros de cada RPC. Se usan dos sintaxis, una más cómoda para programadores en C, y otra para programadores en PASCAL.

El compilador NIDL traduce los comandos NIDL a "subs" ejecutables que serán enlazados con clientes y servidores. Estos "stubs" se generan en código fuente de C pero son totalmente compatibles con los programas de PASCAL.

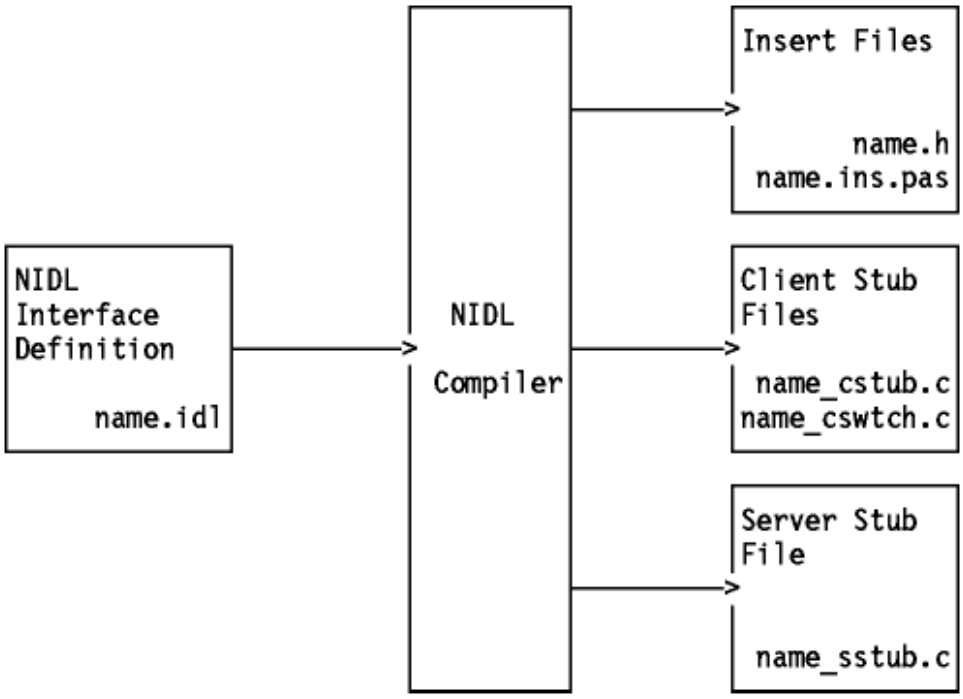


Figura: compilador NIDLr - Ficheros generados.

El compilador NIDL genera dos ficheros "stub" para el cliente: name_cstub.c y name_cswtch.c. El segundo es un fichero que hace de conmutador; se emplea para crear servidores replicados que den acceso a objetos replicados y aseguren consistencia. Las llamadas del cliente se envían al conmutador que contiene los procedimientos públicos, dejando en el "stub" del cliente sólo los privados.

El "stub" generado puede tener dos responsabilidades principales:

- Copiar y convertir datos - Los "tubs" más simples tienen sólo un procedimiento para el paso de argumentos y la conversión de datos. Todos los "stubs" ordenan y recuperan datos en y desde el paquete RPC. El "stub" del cliente ordena los parámetros de entrada para enviar el paquete al servidor y recupera los parámetros de salida del paquete de respuesta. El "stub" del servidor recupera los parámetros de entrada del paquete RPC del cliente y los envía al manager de la interfaz, y ordena los parámetros de salida para enviárselos al cliente. Además, cada "stub" chequea el formato indicado de los datos en el paquete transmitido. Cada sistema envía datos en su formato nativo y el "stub" los convierte a la representación del receptor. Es importante notar que ningún "stub" envía datos en un formato estándar de modo tal que, si ambos sistemas usan la misma representación de los datos, no hay necesidad de convertirlos.
- Liga con una interfaz remota - NIDL gestiona el objeto y la información de la liga de las siguientes formas:
  - Manejador explícito - el cliente pasa explícitamente el manejador como parámetro en cada operación pasada a las rutinas de gestión del servidor.
  - Manejador implícito - el manejador es una sola variable global que hace que el RPC se parezca más a una llamada a un procedimiento corriente aunque la restringe a un sistema con un sólo servidor.
  - Liga manual - el cliente hace todas las llamadas para crear y gestionar el manejador.
  - Liga automática - el cliente llama a una rutina de ligamiento automática para cada RPC, y a una rutina de desligamiento tras recibir la respuesta, cambiando así rendimiento por eficiencia.

4.11.0.3 El Location Broker

El Location Broker es usado por el cliente para solicitar información sobre objetos e interfaces. Los servidores registran esta información en el Location Broker.

El Location Broker se componen de tres elementos:

- Location Broker Local(LLB) - mantiene información sobre objetos e interfaces en el host local y se la suministra a programas de aplicación locales o remotos. También le da al cliente la capacidad de envío del LLB, que evita que un cliente tenga que conocer el puerto específico usado por el servidor.
- Location Broker Global(GLB) - mantiene información sobre objetos e interfaces en toda la red.
- Agente del Cliente del Location Broker - se trata de un conjunto de rutinas llamadas por programas de aplicación para acceder las bases de datos del LLB y el GLB. El cliente puede conocer el host en el que se localiza el objeto e interrogar directamente al LLB del host remoto; de otro modo, tendría que usar el GLB.

El GLB puede tener varias réplicas ejecutándose para asegurar la disponibilidad de la información. Para garantizar la consistencia de los datos de las réplicas, el DRM("Data Replication Manager") se encarga de toda la manipulación, propagando cualquier cambio en la base de datos. El DRM maneja una lista de réplicas que contiene la localización de cada réplica. A los clientes se les permite bloquear el acceso a los objetos y actualizarlos, lo que debilita la consistencia, pero mejora la disponibilidad.

La base de datos del Location Broker tiene los siguientes campos:

- Object UUID - identificador del objeto
- Type UUID - tipo del identificador del objeto
- Interface UUID - interfaz del identificador del objeto
- Flags - indicación de un objeto global
- Annotation - definida por el usuario
- Socket address length - longitud de la dirección del zócalo
- Socket address - localización del servidor

Aquí hay algunas definiciones:

- *UUID* significa *Identificador Universal Unívoco ("Universal Unique IDentifier")*, es decir, un valor de 128 bits usado para la identificación. Ningún otro objeto, tipo o interfaz puede hacer uso de un UUID que ya ha sido asignado.
- *Objeto*: entidad manipulada por operaciones bien conocida. Ficheros de disco e impresoras son ejemplos de objetos. Las interfaces son el medio de acceso a los objetos. Todo objeto tiene un tipo.
- *Tipo*: un clase de objeto. Se puede acceder a todos los objetos del mismo tipo específico a través de la misma interfaz o interfaces.
- *Interfaz*: conjunto de operaciones. El NCA especifica un Lenguaje de Definición de Interfaz de Red("Network Interface Definition Language(NIDL)") para definir interfaces.

- *NIDL*: lenguaje declarativo para la definición de interfaces.
- *Compilador NIDL*: herramienta NCS que convierte la definición de una interfaz, escrita en NIDL, en varios módulos de programa, que incluyen código fuente para los "stubs" del cliente y del servidor.

Tabla de contenidos NFS ("Network File System(NFS)")

4.12 NFS("Network File System")

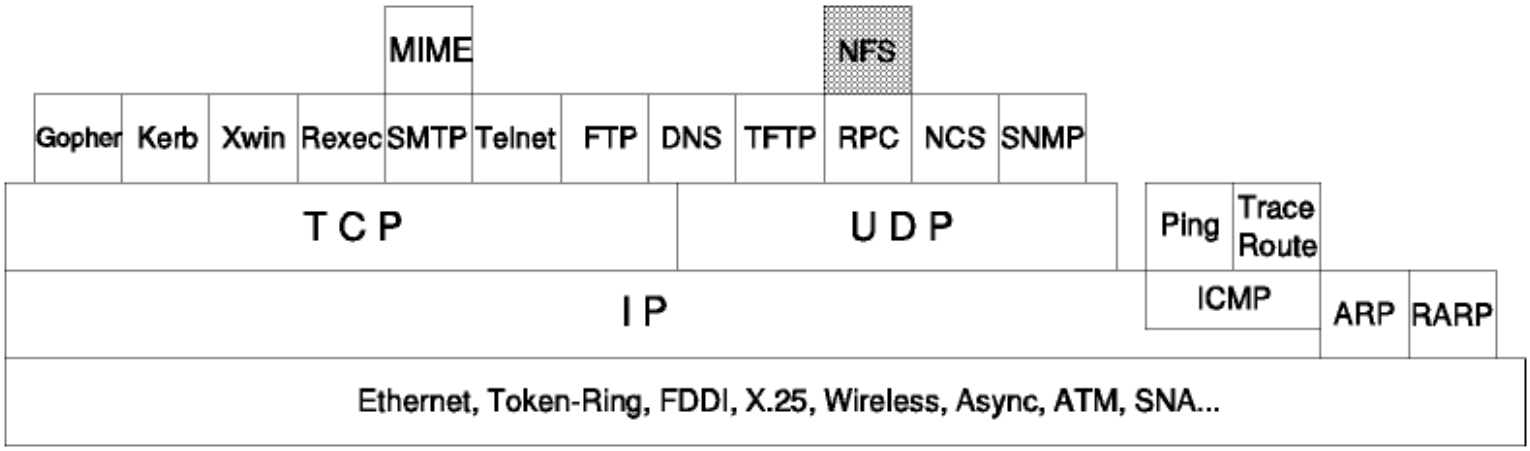


Figure: NFS("Network File System")

El protocolo NFS("Network File System") de Sun permite compartir sistemas de ficheros en una red. El protocolo NFS está diseñado para ser independiente de la máquina, el sistema operativo y el protocolo de transporte. Esto es posible porque se implementa sobre RPC(ver RPC("Remote Procedure Call")). RPC establece la independencia de la máquina usando XDR("External Data Representation").

SUN-NFS es un *protocolo propuesto como estándar*. Su status es *electivo*. La especificación actual del NFS de Sun se puede encontrar en el *RFC 1094 - especificación de NFS:*. Este RFC documenta la versión 2 de NFS. Aunque el RFC menciona a la versión 3 de NFS, nadie ha enviado aún un nuevo RFC ni un borrador discutiendo una posible especificación de la versión 3.

4.12.1 Concepto

NFS permite a los usuarios autorizado acceder a ficheros localizados en sistemas remotos como si fueran locales. Dos protocolos sirven a este propósito:

1. El protocolo de montaje *Mount* para especificar el host remoto y el sistema de ficheros al que se va a acceder así como dónde localizarlo en la jerarquía local de ficheros.
2. El protocolo *NFS* para efectuar la auténtica E/S de ficheros sobre el sistema remoto de ficheros.

Tanto el protocolo Mount como el NFS son aplicaciones de RPC(concepto de llamador/servidor) y son *transportados por UDP*.

4.12.1.1 Protocolo

El protocolo Mount es una aplicación de RPC integrada con NFS. Su número de programa es el 100005. El protocolo Mount es transportado por UDP. Mount es un servidor RPC y proporciona un total de seis procedimientos:

NULL
: No hace nada, es útil para testear las respuestas del servidor

MOUNT
: Función Mount, devuelve un descriptor de fichero apuntando al directorio

DUMP
: Devuelve la lista de todos los sistemas de ficheros montados

UMOUNT
: Elimina una entrada de la lista de sistemas de ficheros montados

UNMTALL
: Elimina todas de las entradas de la lista de sistemas de ficheros montados para el cliente

EXPORT
: Devuelve información sobre los sistemas de ficheros disponibles

La llamada a *MOUNT* devuelve un descriptor de fichero que apunta al directorio. Este descriptor es un campo de 32 bytes, que el cliente usará posteriormente para acceder a los ficheros. Los descriptores son una parte fundamental de NFS ya que a través de ellos se referenciará cada fichero y directorio. Algunas implementaciones encriptan los descriptores por razones de seguridad(por ejemplo, el NFS de VM puede usar opcionalmente programas de encriptación para esto).

El comando MOUNT aporta la interfaz a esta aplicación de RPC. El usuario ejecuta el comando MOUNT para localizar el sistema de ficheros remoto en su propia jerarquía de ficheros.

4.12.1.2 Protocolo NFS

NFS es la aplicación de RPC que suministra funciones de E/S sobre ficheros en un host remoto, una vez que ha sido solicitado mediante el comando MOUNT. Tiene el número de programa 100003 y a veces usa el puerto IP 2049. Como este no es un número de puerto asignado oficialmente, y ya existen diversas versiones de NFS(y de MOUNT) lo números de puerto pueden cambiar. Es aconsejable dirigirse al Portmap(número de puerto 111) (ver Portmap o Portmapper) para obtener los números de puerto para MOUNT y NFS. El protocolo NFS es transportado por UDP.

El programa NFS soporta 18 procedimientos para todas las operaciones básicas de E/S, como por ejemplo:

LOOKUP
: Busca un fichero en el directorio actual y si lo encuentra, devuelve un descriptor a ese fichero más información sobre los atributos del fichero.

READ y WRITE
: Primitivas básicas para acceder el fichero.

RENAME
: Renombra un fichero.

REMOVE
: Borra un fichero.

MKDIR y RMDIR
: Creación/borrado de subdirectorios.

GET y SET-ATTR
: Devuelve conjuntos de atributos de ficheros.

Además hay otras funciones soportadas.

Se correponden con la mayoría de primitivas de E/S usadas en el sistema operativo local para acceder a ficheros locales. De hecho, una vez que se ha montado el directorio remoto, el sistema operativo local tiene que "reencaminar" las primitivas de E/S al host remoto. Esto hace que todas las operaciones de E/S sobre ficheros tengan el mismo aspecto, independientemente de si el fichero es local o remoto. El usuario puede trabajar con los comandos y programas habituales en ambos tipos de ficheros; en otras palabras, el *protocolo NFS es completamente transparente al usuario*. Ver Figura - E/S de ficheros en NFS.

Figura: E/S de ficheros en NFS - Interceptada a nivel de sistema operativo, haciéndose por tanto transparente al usuario.

4.12.1.3 Sistema de ficheros NFS

NFS asume un sistema de ficheros jerárquico(directorios). Los ficheros son secuencias de bytes sin estructura y carentes de significado inherente: es decir, todos lo ficheros se ven como una secuencia contigua de bytes, sin ninguna estructura de registros.

Con NFS, todas las operaciones sobre ficheros son *síncronas*. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará a estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los ficheros.

NFS también especifica que los servidores no deberían estar *asociados a un cliente determinado.* Es decir, un servidor no necesita mantener información extra de ninguno de sus clientes para funcionar correctamente. En caso de un fallo del servidor, los clientes sólo tienen que reintentar la solicitud hasta que el servidor responda, sin tener que repetir la operación de mount.

Tabla de contenidos El sistema de autentificación y autorización Kerberos

4.13 El sistema de autentificación y autorización Kerberos

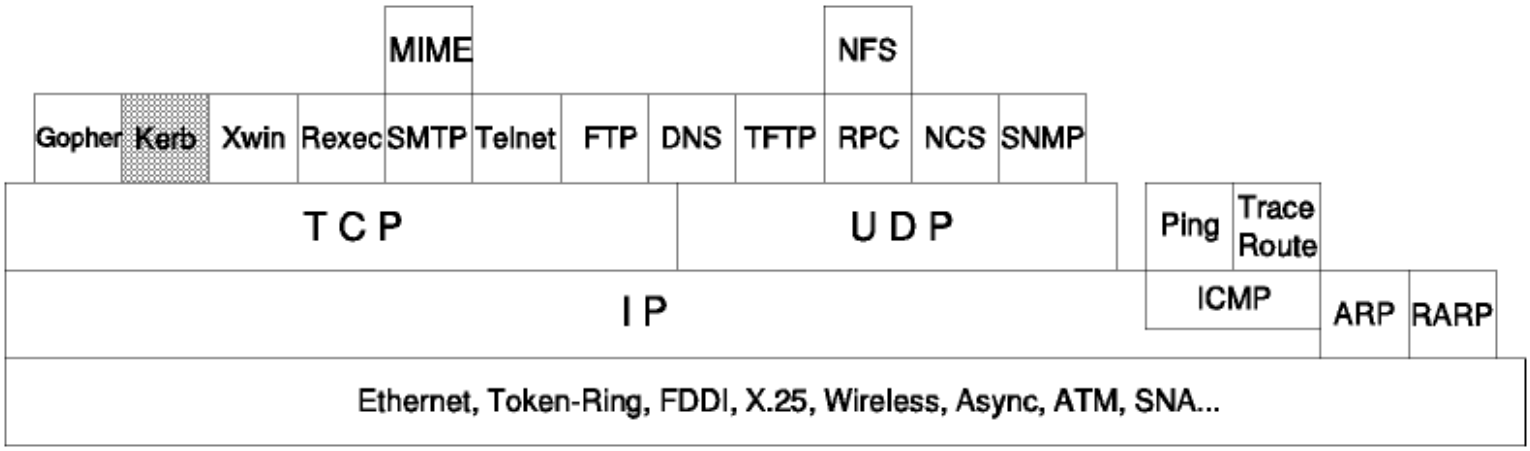


Figura: El sistema de autentificación y autorización Kerberos

Según el *Gran Diccionario del Diablo("Enlarged Devil's Dictionary(Ambrose Bierce)")*, Cerbero es "el perro guardián del Hades, cuyo deber era guardar la entrada del infierno; se sabe que Cerbero tenía tres cabezas".

El sistema de autentificación y autorización Kerberos es un sistema de seguridad basado en la encriptación que proporcionar autentificación mutua entre usuarios y servidores en un entorno de red. Las metas asumidas por este sistema son:

- Autentificación para evitar solicitudes/respuestas fraudulentas entre servidores y usuarios que deben tener índole confidencial y en grupos de al menos un usuario y un servicio.
- Cada servicio que desee proporcionar su propio sistema de autorización puede implementarlo independientemente de la autentificación. El sistema de autorización puede asumir que el sistema de autentificación usuario/cliente es fiable.
- Permitir la implementación de un sistema de contabilidad que esté integrado y sea seguro y fiable, con estructura modular y soporte para facturación

El sistema Kerberos se usa principalmente con propósitos de autentificación, aunque también aporta la flexibilidad necesaria para añadir información de auto.

Las versiones actual del protocolo Kerberos son las 4 y 5. La versión 4 tiene un amplio uso y es la que se suele implementar en productos comerciales. La versión 5 está propuesta actualmente como borrador de Internet. Se basa en la versión 4 e incorpora cierto número de mejoras y nuevas características.

4.13.1 Supuestos

Kerberos da por hecho lo siguiente:

- El entorno que lo usará incluirá: estaciones de trabajo públicas y privadas que estarán localizadas en áreas con seguridad física mínima; una red universitaria sin encriptación de los enlaces que puede estar compuesta de redes locales dispersas conectadas por troncales o pasarelas; servidores operados de modo centralizado en habitaciones cerradas con seguridad física moderada o alta.
- Los datos confidenciales o las operaciones de alto riesgo tales como transacciones bancarias no pueden formar parte de este entorno sin seguridad adicional, ya que una vez que tienes una estación de trabajo como terminal puedes emular ciertas condiciones y los datos normales fluirán sin ningún tipo de protección por encriptación.
- Uno de los sistemas de encriptación utilizados es el DES("Data Encryption Standard"), que está disponible en el mercado de U.S pero no puede ser exportado sin una licencia oficial de exportación, por lo que los diseñadores de Kerberos lo han desarrollado para que sea modular y admita recambios o sustituciones.
- Kerberos asume la existencia de un reloj débilmente sincronizado en todo el sistema así que la estación de trabajo dispone de una herramienta de sincronización como la que le suministre el servidor de sincronización("Time server").

4.13.2 Nombres

Un *Identificador Principal* es el nombre que define un cliente o un servicio en el sistema Kerberos.

En la versión 4, el identificador consta de tres componentes:

- El nombre *principal* es único para cada cliente y servicio asignado por el Manager o Administrados de Kerberos.
- El nombre de *instancia*, usado para la autentificación es una etiqueta añadida para clientes y servicios que existe bajo diversas formas. Para usuarios, un instancia puede proporcionar diferentes instancias para máquinas diferentes. Para servicios, una instancia suele especificar el nombre del host en la máquina que proporciona el servicio.
- El nombre de *reino*, usado para permitir sitios administrados independientemente con Kerberos. El nombre principal y el de instancia son calificados por el del reino al que pertenecen, y son únicos sólo dentro de ese reino. El reino es habitualmente el nombre del dominio.

En la versión 4, cada uno de estos componentes tiene un límite de longitud de 39 caracteres. Debido a convenios, (.) no es un carácter aceptado.

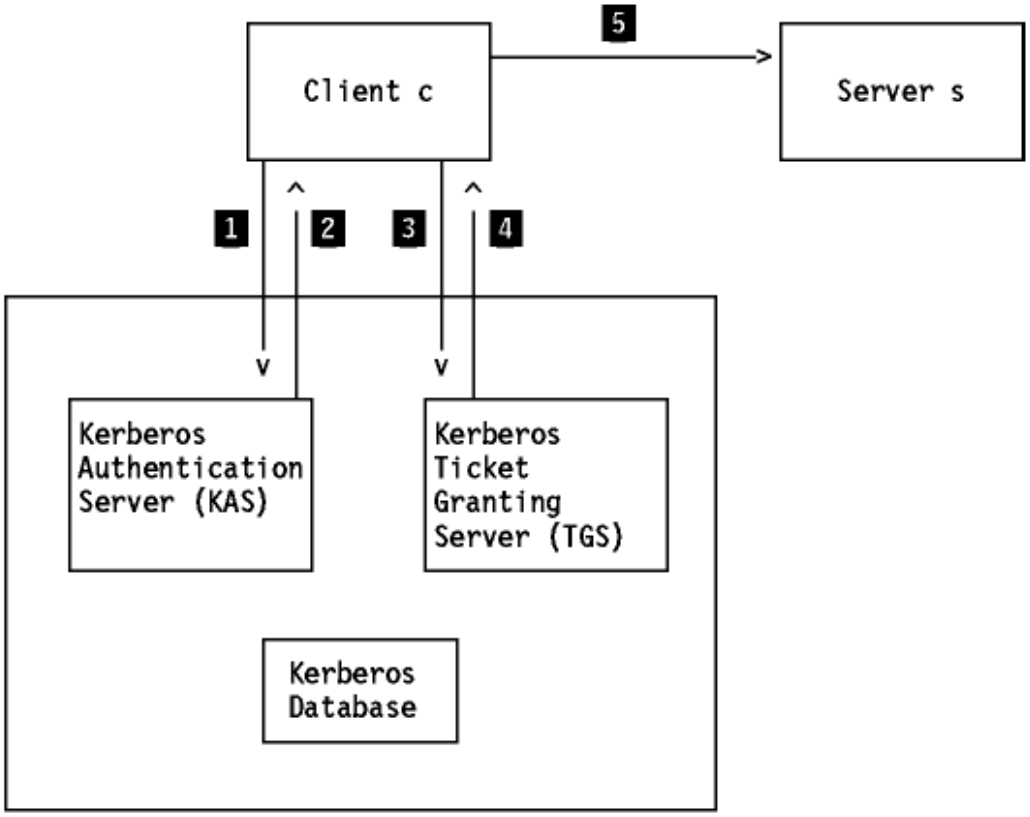
En la versión 5, el identificador consta sólo de dos partes, el *reino* y el *resto*, que es una secuencia de cuantos componentes sean necesarios para determinar al principal. Tanto el reino como cada componente del resto se definen como *Ristras Genéricas("GeneralStrings")* de ASN.1("Abstract Syntax Notation One", estándar ISO 8824), lo que supone pocas restricciones para los caracteres disponibles para los identificadores principales.

4.13.3 Proceso de autentificación en Kerberos

En el sistema Kerberos, el cliente que desea contactar con un servidor para que le dé un servicio, debe pedir primero un *ticket* de una tercera parte de mutua confianza, el **KAS**("Kerberos Authentication Server"). Este ticket se obtiene como una función en la que uno de los componentes es una llave privada conocida sólo por el servicio y el KAS, de modo que el servicio puede estar seguro de que el ticket procede sólo de Kerberos. El KAS conoce al cliente por su nombre principal(**c**). La llave privada(**K(c)**) es la llave de

autentificación conocida sólo por el usuario y el KAS.

En este capítulo, el símbolo **{X,Y}** indica un aje que contiene información o datos X e Y. **{X,Y}K(z)** indica que un mensaje que contiene los datos X e Y ha sido cifrado usando la llave k(z).



1 Client —> KAS: c, tgs, n
2 KAS —> Client: $\{K_{c,tgs}, n\}K_c$, $\{T_{c,tgs}\}K_{tgs}$
3 Client —> TGS: $\{A_c\}K_{c,tgs}$, $\{T_{c,tgs}\}K_{tgs}$, s, n
4 TGS —> Client: $\{K_{c,s},n\}K_{c,tgs}$, $\{T_{c,s}\}K_s$
5 Client —> Server: $\{A_cn\}K_{c,s}$, $\{T_{c,s}\}K_s$

In Kerberos Version 4:

message 2 was: KAS —> Client: $\{K_{c,tgs},n,\{T_{c,tgs}\}K_{tgs}\}K_c$
message 4 was: TGS —> Client: $\{K_{c,s},n,\{T_{c,s}\}K_s\}K_{c,tgs}$

Figura: Esquema de autentificación de Kerberos

El proceso de autentificación consiste en el intercambio de cinco mensajes(ver Figura: Esquema de autentificación de Kerberos):

1 Cliente -> KAS

El cliente envía un mensaje {c, tgs, n}, al KAS, que contiene su identidad(c), un una palabra temporal o "nonce"(un sello de tiempo u otro forma de identificar su solicitud), y solicita un ticket para usarlo con el servidor de tickets(TGS).

2 KAS -> Cliente

El servidor de autentificación busca el nombre del cliente(c) y el nombre del servicio(TGS, el servidor de tickets) en la base de datos de Kerberos, y obtiene una llave de encriptación para cada uno de ellos K(c) y K(TGS).

El KAS crea a continuación una respuesta para enviársela al cliente. Esta respuesta contiene un ticket inicial T(c, TGS) que garantiza al cliente acceso al servidor solicitado(el TGS). T(c, TGS) contiene k(c.TGS), c, TGS, el "nonce", el tiempo de vida y otra información. El KAS también genera una llave aleatoria de encriptación K(c, TGS), llamada llave de sesión. Luego encripta este ticket usando la llave de encriptación del TGS(K(TGS)). Este procedimiento es lo que se denomina un *ticket sellado* {T(c,tgs)}K(tgs). Inmediatamente se crea un mensaje consistente en el ticket sellado y la llave de sesión de TGS K(c, TGS).

Nota: En la versión 4 de Kerberos, el mensaje es:
{K(c,tgs),n,{T(c,tgs)}K(tgs)}K(c)
, mientras que en la 5 el mensaje tiene una forma más simple:
{K(c,tgs), n}K(c), {T(c,tgs)}K(tgs)
Esto simplifica la doble encriptación(innecesaria) del ticket.

3 Cliente -> TGS

A la recepción del mensaje, el cliente lo desencripta usando su llave secreta K(c) que es la única que él y el KAS conocen. Comprueba que el "nonce" (n) coincide con la solicitud específica, y entonces guarda la llave de sesión K(c,tgs) para futuras comunicaciones con el TGS.

El cliente envía a continuación un mensaje al YGS. Este mensaje contiene el ticket inicial {T(c,tgs)}K(tgs), el nombre del servidor(s), un "nonce", y un nuevo autentificador A(c) que lleva un sello de tiempo. A(c) es {c, nonce}. El mensaje es:
{A(c)}K(c,tgs), {T(c,TGS)}K(TGS), s, n

4 TGS -> Cliente

El TGS recibe el mensaje anterior del cliente(c), y descifra primero el ticket sellado usando su llave de encriptación TGS(este ticket fue sellado originalmente por el KAS en el paso 2 usando la misma llave). El TGS obtiene la llave de sesión para TGS de del ticket descifrado, y la emplea a su vez para descifrar el autentificador sellado (la validez se chequea comparando el nombre del cliente tanto en el ticket como en el autentificador, el nombre del servidor TGS que figura en el ticket, la dirección de red que debe ser igual en el ticket, en el autentificador y en el mensaje recibido). Finalmente, chequea la hora actual en el autentificador para cerciorarse de que el mensaje es reciente. *Esto requiere que todos los clientes y servidores matengan sus relojes dentro de cierto margen prescrito de tolerancia*. Ahora el TGS busca el nombre del servidor que aparece en el mensaje en la base de datos de Kerberos, y obtiene la lave de encriptación(K(s)) para el servicio especificado.

El TGS crea una nueva llave aleatoria de sesión K(c,s) para el cliente(c) y el servidor, para generar a continuación un nuevo ticket. T(c,s) que contiene:
K(c,s), n, "nonce", tiempo de vida

Luego ensambla un mensaje y lo envía al cliente.

Nota: En la versión 4 de Kerberos Version, el mensaje es:
{K(c,s),n,{T(c,s)}K(s)}K(c,tgs)
En la versión 5 el mensaje tiene una forma más simple:
{K(c,s),n}K(c,tgs), {T(c,s)}K(s)
Esto simplifica la doble encriptación(innecesaria) del ticket.

5 Cliente -> Servidor

El cliente recibe este mensaje y lo descifra usando la llave de sesión para el TGS que sólo comparten él y el TGS. De este mensaje calcula una nueva llave de sesión K(c,s) que comparte con el servidor(s) y un ticket sellado que no puede descifrar porque está cifrado con la llave secreta del servidor K(s).

El cliente construye un autentificador y lo sella con la nueva llave de sesión K(c,s). Por último, envía un mensaje que contiene el ticket sellado y el autentificador al servidor(s) para solicitar su servicio.

El servidor(s) recibe este mensaje y descifra primero el ticket sellado con su llave de encriptación, conocida sólo por él y el KAS. Luego usa la nueva llave de sesión contenida en el ticket para descifrar el autentificador y realiza el mismo proceso de validación que el descrito en el paso 4.

Una vez que el servidor ha validado a un cliente, el cliente tiene la opción de validar a su vez al servidor. Esto evita que un intruso suplante al servidor. El cliente requiere que el servidor le devuelva un mensaje con el sello de tiempo(procedente del autentificador del cliente, incrementado en uno). Este mensaje se cifra con la llave de sesión que pasó del cliente al servidor.

Resumamos los puntos centrales de este esquema:

- Con el fin de que la estación de trabajo emplee cualquier servidor, se requiere un ticket. Todos los tickets, a excepción del primero(también llamado *ticket inicial*) se obtienen del TGS. El primer ticket es especial: es un ticket para el propio TGS y se obtiene del KAS.
- Cada ticket está asociado con una llave de sesión que se asigna cada vez que se concede un ticket.
- Los tickets son reutilizables. Cada ticket tiene un tiempo de vida, típicamente de ocho horas. Después de que un ticket ha expirado, el usuario ha de identificarse de nuevo al Kerberos, introduciendo el nombre de usuario y el password.
- A diferencia de un ticket, que puede ser reutilizado, hace falta un nuevo autentificador cada vez que el cliente inicia una conexión con el servidor. El autentificador contiene un sello de tiempo que expira a los pocos minutos de haber sido expedido(esta es la razón por la que los relojes de clientes y servidores deben estar sincronizados).
- Un servidor debería mantener un seguimiento de las solicitudes anteriores de los clientes para las que el sello de tiempo en el identificador es aún válido. De este modo el servidor puede rechazar solicitudes duplicadas que podrían surgir de un ticket y un identificador robados.

4.13.4 Administración de la base de datos de Kerberos

Kerberos necesita un registro para cada usuario y servicio de su reino y cada registro sólo contiene la información necesaria, que es la siguiente:

- Identificador principal(c,s)
- Llave privada para el identificador principal(K(c),K(s))
- Fecha de expiración de la identidad
- Fecha de la última modificación del registro
- Identidad de la persona que modificó por última vez el registro(c,s)
- Tiempo de vida de los tickets concedidos al identificador principal
- Atributos(no usado)
- Implementación de datos(no visible externamente)

El campo de llave privada se cifra con una llave maestra de forma que la eliminación de la base de datos no causará ningún problema puesto que la llave maestra no está en ella.

La entidad responsable de gestionar la base de datos es el KDBM("Kerberos Database Manager"). Sólo hay un KDBM en un reino, pero es posible tener más de un KKDS("Kerberos Key Distribution Server"), cada uno con una copia de la base de datos de Kerberos. Esto se hace para mejorar la disponibilidad y el rendimiento de modo que el usuario pueda elegir de entre un grupo de KKDSs a cuál enviar su petición. El KKDS efectúa sólo operaciones de lectura, dejando la actualización al KDBM, que copia toda la base de datos unas cuantas veces al día. Así se simplifica la operación al usar un protocolo protegido de Kerberos. Este protocolo es básicamente una autentificación mutua entre el KDBM Y EL KDDS seguida de una operación de transferencia de dicheros con campos de checksum y checkpoint.

4.13.5 Modelo de autorización de Kerberos

El modelo de autentificación de Kerberos sólo permite al servicio verificar la identidad del solicitante pero no da información de si éste puede usar o no el servicio. El lema de autorización de Kerberos se basa en el principio de que cada servicio conoce al usuario de modo que puede mantener su propia información de autorización. Sin embargo, el sistema de autentificación de Kerberos se podría extender con información y algoritmos que servirían para propósitos de autorización. (Esto es más fácil en la versión 5. Ver la siguiente sección). El Kerberos podría chequear entonces si un usuuario/cliente está autorizado a usar cierto servicio.

Obviamente, tanto la aplicación del cliente como la del servidor deben ser capaces de manejar el proceso de autentificación de Kerberos. Es decir, cliente y servidor deben estar *kerberizados*.

4.13.6 Mejoras de la versión 5 de Kerberos

La versión 5 de Kerberos tiene una serie de mejoras sobre la versión 4. Algunas de las importantes son:

- El uso de la encriptación se ha separado en distintos módulos de programa que permiten soporta múltiples sistemas de encriptado.
- Las direcciones de red que aparecen en mensajes de protocolo se marcan con campos de longitud y tipo. Esto permite la compatibilidad con múltiples protocolos de red.
- La codificación de mensajes se describe con la sintaxis ASN.1("Abstract Syntax Notation 1") de acuerdo con los estándares ISO 8824 y 8825.
- El ticket en la versión 5 de Kerberos tiene un formato expandido para soportar nuevas características(por ejemplo, la cooperación entre reinos).
- Como se menciona en Nombres, el identificador principal ha cambiado.
- Se ha mejorado el soporte a operaciones entre reinos.
- La información de autorización y contabilidad ya se puede encriptar y transmitir en un ticket con el campo de autorización de datos. Esto facilita la extensión del esquema de autentificación para que incluya también un esquema de autorización.
- La implementación e la versión 5 de Kerberos proporcionar una liga con el GSSAPI("Generic Security Service API").

Para más detalles sobre la diferencia entre las dos versiones de Kerberos, remitirse a [Kohl, Neuman y Ts'o]. Para detalles sobre GSSAPI, remitirse a [Linn].

 Tabla de contenidos  Gestión de redes

4.14 Gestión de red

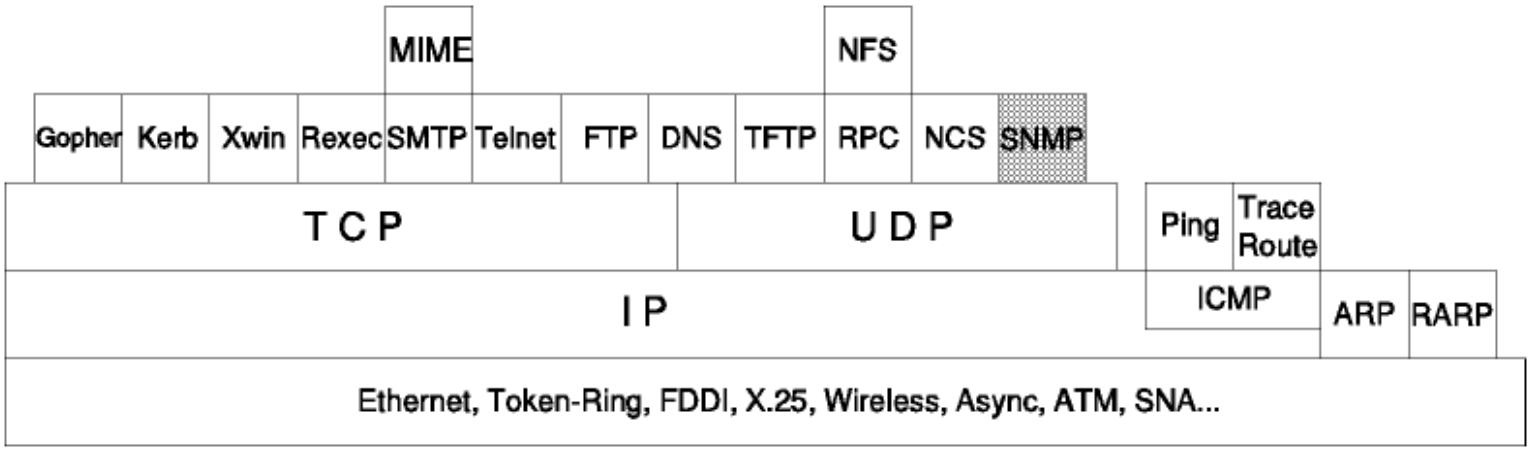


Figura: Gestión de red

Con el crecimiento en tamaño y complejidad de las redes basadas en TCP/IP, la necesidad de mecanismos de gestión de red se ha vuelto muy importante. En la actualidad, los protocolos que forman el soporte de la gestión de red son:

1. SMI (RFC 1155) - describe cómo se definen los objetos gestionados contenidos en el MIB (se verá en SMI("Structure and Identification of Management Information").)
2. MIB-II (RFC 1213) - describe los objetos gestionados contenidos en el MIB(se verá en MIB("Management Information Base").)
3. SNMP (RFC 1098) - define el protocolo usado para gestionar estos objetos(se verá en SNMP("Simple Network Management Protocol").)

El IAB("Internet Architecture Board") emitió un RFC al respecto, en el que adoptaba dos actitudes diferentes:

- A corto plazo, se recomienda el uso de SNMP.

  El IAB recomienda que todas las implementaciones de IP y TCP sean gestionables. Actualmente, esto implica la implementación de MIB-II (RFC 1213), y de al menos el protocolo recomendado de gestión SNMP (RFC 1157).

  Notar que los protocolos históricos SGMP (*Simple Gateway Monitoring Protocol*, RFC 1028) y MIB-I (RFC-1156) no están recomendados.

- A largo plazo, se debería investigar el uso del incipiente protocolo de gestión de red de OSI(CMIP). Es lo que se conoce como *CMIP sobre TCP/IP* (CMOT).(Se discutirá en CMOT ("Common Management Information Protocol over TCP/IP")).

Tanto SNMP y CMOT utilizan los mismos conceptos básicos en la descripción y definición la información de gestión denominada *SMI("Structure and Identification of Management Information")* descrita en el RFC 1155 y *MIB("Management Information Base")*, descrita en el RFC 1156.

4.14.1 Estándares

SNMP (*Simple Network Management Protocol*) es un *protocolo estándar* de Internet. Su status es *recomendado*. Su especificación actual se encuentra en el *RFC 1157 - SNMP("Simple Network Management Protocol")*.

MIB-II es un *protocolo estándar* de Internet. Su status es *recomendado*. Su especificación actual se encuentra en el *RFC 1213 - MIB-II:Management Information Base for Network Management of TCP/IP-based Internets*.

CMIP (*Common Management Information Protocol*) y CMIS (*Common Management Information Services*) se definen según los estándares ISO/IEC 9595 y 9596.

CMOT (*CMIS/CMIP Over TCP/IP*) es un *protocolo propuesto como estándar de Internet*. Su status es *electivo*. Su especificación actual se encuentra en el *RFC 1189 - CMOT y CMIP("Common Management Information Services and Protocols for the Internet")*.

OIM-MIB-II es un *protocolo propuesto como estándar de Internet*. Su status es *electivo*. Su especificación actual se encuentra en el *RFC 1214 - Gestión OSI Internet Management: MIB("Management Information Base")*.

Otros RFCs emitidos por el IAB sobre este tema son:

- RFC 1052 - *Recomendaciones del IAB para el desarrollo de estándares de gestión de red*.
- RFC 1085 - *Servicios ISO de presentación sobre redes basadas en TCP/IP*.
- RFC 1155 - *SMI("Structure and Identification of Management Information") para redes basadas en TCP/IP*.
- RFC 1156 - *MIB("Management Information Base") para la gestión de redes basadas en TCP/IP*.
- RFC 1215 - *Convenios de definición para SNMP*.
- RFC 1227 - *Protocolo SNMP MUX y MIB*.
- RFC 1228 - *SNMP-DPI("Simple Network Management Protocol Distributed Programming Interface"*.
- RFC 1230 - *IEEE 802.4 Token Bus MIB*.
- RFC 1231 - *IEEE 802.5 Token-Ring MIB*.
- RFC 1239 - *Reasignación de MIBs experimentales a MIBs estándares*.
- RFC 1351 - *Modelo administrativo de SNMP*.
- RFC 1352 - *Protocolos de seguridad SNMP*.

4.14.2 SMI("Structure and Identification of Management Information")

El SMI define las reglas para describir los objetos gestionados y cómo los protocolos sometidos a la gestión pueden acceder a ellos. La descripción de los objetos gestionados se hace utilizando un subconjunto de ASN.1("Abstract Syntax Notation 1, estándar ISO 8824), un lenguaje de descripción de datos. La definición del tipo de objeto consta de cinco campos:

- Objeto: nombre textual, llamado *descriptor del objeto*, para el tipo del objeto, junto con su correspondiente *identificador de objeto*, definido abajo.
- Sintaxis: la sintaxis abstracta para el tipo el objeto. Las opciones son SimpleSyntax (entero, octeto de caracteres, identificador de objeto, Null), ApplicationSyntax (dirección de red, contador, escala, ticks, opaco) u otro tipo de sintaxis de aplicación(ver el RFC 1155 para más detalles).
- Definición: descripción textual de la semántica del tipo.
- Acceso: sólo lectura, sólo escritura, lectura - escritura o inaccesible.
- Status: obligatorio, opcional u obsoleto.

Como ejemplo, podemos tener:

```
OBJECT
        sysDescr { system 1 }
Syntax  OCTET STRING
Definition This value should include the full name and version
           identification of the system's hardware type, software
           operating-system, and networking software. It is
           mandatory that this contain only printable ASCII
           characters.
Access  read-only.
Status  mandatory.
```

Este ejemplo muestra la definición de un objeto contenido en el MIB. Su nombre es sysDescr y pertenece al grupo sistema(ver MIB("Management Information Base")).

Un objeto gestionado no sólo ha de ser descrito, también debe ser identificado. Esto se hace utilizando el identificado de objeto("Object Identifier") ASN.1 como si fuera un número de teléfono, reservando grupos de números para distintas localizaciones. En el caso de la gestión de red para TCP/IP, el número reservado fue 1.3.6.1.2 y SMI lo usa como base para la definición de nuevos objetos.

Este número se obtiene al unir a grupos de números con el siguiente significado:

- El primer grupo define el nodo administrador:
  - (1) para ISO
  - (2) para CCITT
  - (3) para la unión ISO-CCITT.
- El segundo grupo para el nodo administrador ISO define (3) para su uso por parte de otras organizaciones.
- El tercer grupo define (6) para su uso por parte del DoS("U.S. Department of Defense").
- En el cuarto grupo, el DoD no ha indicado cómo ha de gestionarse se grupo correspondiente por lo que la comunidad de Internet ha asumido (1).
- El quinto grupo fue aprobado por el IAB para ser:
  - (1) para el uso del directorio OSI en Internet
  - (2) para la identificación de objetos con propósitos de gestión
  - (3) para la identificación de objetos con fines experimentales
  - (4) para la identificación de objetos para uso privado

En el ejemplo, {system 1} significa que el identificador del objeto es 1.3.6.1.2.1.1.1. Es el primer objeto en el primer grupo(sistema) en el MIB.

4.14.3 MIB("Management Information Base")

4.14.3.1 Descripción

El MIB define los objetos que pueden ser gestionados para cada capa en el protocolo TCP/IP. Hay dos versiones, MIB-I and MIB-II. MIB-I fue definida en el RFC 1156, y está clasificado ahora como protocolo *histórico* con status *no recomendado*.

Group	Objects for	#
system	basic system information	7
interfaces	network attachments	23
at	address translation	3
ip	internet protocol	38
icmp	internal control message protocol statistics	26
tcp	transmission control protocol	19
udp	user datagram protocol	7
egp	exterior gateway protocol	18
transmiss.	transmission. Media-specific	0
snmp	snmp applications entities	30

#: Number of objects in the group

Figura: MIB - II("Management Information Base II") - Definición de grupo.

Cada nodo gestionado soporta sólo los grupos apropiados. Por ejemplo, si no hay pasarela, el grupo EGP no tiene por qué estar incluido. Pero si un grupo es apropiado, todos los objetos en ese grupo deben estar soportados.

La lista de objetos gestionados definidos deriva de aquellos elementos considerados esenciales. Este enfoque, consistente en tomar sólo los objetos esenciales no es restrictivo, ya que el SMI proporciona mecanismos de extensibilidad tales como la definición de una nueva versión de MIB o de objetos privados o no estandarizados.

Debajo hay algunos ejemplos de objetos de cada grupo. La lista completa está definida en el RFC 1213.

- Grupo de sistema
  - sysDescr - Descripción completa del sistema(version, HW, OS)
  - sysObjectID - Identificación que da el distribuidor al objeto
  - sysUpTime - Tiempo desde la última reinicialización
  - sysContact - Nombre de la persona que hace de contacto
  - sysServices - Servicios que ofrece el dispositivo
- Grupo de interfaces
  - ifIndex - Número de interfaz
  - ifDescr - Descripción de la interfaz
  - ifType - Tipo de la interfaz
  - ifMtu - Tamaño máximo del datagrama IP
  - ifAdminisStatus - Status de la interfaz
  - ifLastChange - Tiempo que lleva la interfaz en el estado actual
  - ifINErrors - Número de paquetes recibidos que contenían errores
  - ifOutDiscards - Número de paquetes enviados y desechados
- Grupo de traducción de direcciones
  - atTable - Tabla de traducción de direcciones
  - atEntry - Cada entrada que contiene una correspondencia de dirección de red a dirección física
  - atPhysAddress - La dirección física dependiente del medio
  - atNetAddress - La dirección de red correspondiente a la dirección física
- Grupo IP
  - ipForwarding - Indicación de si la entidad es una pasarela IP
  - ipInHdrErrors - Número de datagramas de entrada desechados debido a errores en sus cabeceras IP
  - ipInAddrErrors - Número de datagramas de entrada desechados debido a errores en sus direcciones IP
  - ipInUnknownProtos - Número de datagramas de entrada desechados debido a protocolos desconocidos o no soportados
  - ipReasmOKs - Número de datagramas IP reensamblados con éxito
  - ipRouteMask - Máscara de subred para el encaminamiento
- Grupo ICMP
  - icmpInMsgs - Número de mensajes ICMP recibidos
  - icmpInDestUnreachs - Número de mensajes ICMP "destino inalcanzable"(destination unreachable) recibidos
  - icmpInTimeExcds - Número de mensajes ICMP "time exceeded"(tiempo excedido) recibidos
  - icmpInSrcQuenchs - Número de mensajes ICMP "source quench(desbordamiento del emisor) recibidos
  - icmpOutErrors - Número de mensajes ICMP no enviados debido a problemas en ICMP
- Grupo TCP
  - tcpRtoAlgorithm - Algoritmo que determina el timeout para retransmitir octetos para los que no se ha recibido reconocimiento
  - tcpMaxConn - Límite en el número de conexiones TCP que puede soportar la entidad
  - tcpActiveOpens - Número de veces que las conexiones TCP han efectuado una transición directa del estado SYN-SENT al estado CLOSED
  - tcpInSegs - Número de segmentos recibidos, incluyendo aquellos con error
  - tcpConnRemAddress - La dirección IP remota para esta conexión TCP
  - tcpInErrs - Número de segmentos desechados debido a errores de formato
  - tcpOutRsts - Número de resets generados
- Grupo UDP
  - udpInDatagrams - Número de datagramas UDP entregados a usuarios UDP
  - udpNoPorts - Número de datagramas UDP recibidos para los que no existía aplicación en el puerto de destino
  - udpInErrors - Número de datagramas UDP recibidos que no se pudieron entregar por razones otras que la ausencia de la aplicación en el puerto de destino
  - udpOutDatagrams - Número de datagramas UDP enviados por la entidad
- Grupo EGP
  - egpInMsgs - Número de mensajes EGP recibidos sin error
  - egpInErrors - Número de mensajes EGP con error
  - egpOutMsgs - Número de mensajes EGP generados localmente
  - egpNeighAddr - La dirección IP del vecino de esta entrada EGP
  - egpNeighState - El estado EGP del sistema local con respecto a la entrada EGP vecino

Esta no es la definición completa del MIB pero sirve de ejemplo de los objetos de cada grupo.

El grupo de interfaces contiene dos objetos de nivel superior: el número de interfaces del nodo(*ifNumber*) y una tabla con información de estas(*ifTable*). Cada entrada de la tabla(*ifEntry*) contiene los objetos de esa interfaz. Entre ellos, el tipo de interfaz(*ifType*) se identifica en el árbol MIB con notación ASN.1 como 1.3.6.1.2.1.2.2.1.3. Para un adaptador de red en anillo, su valor sería 9("iso88025-tokenRing") (ver Figura - Identificador de objeto).

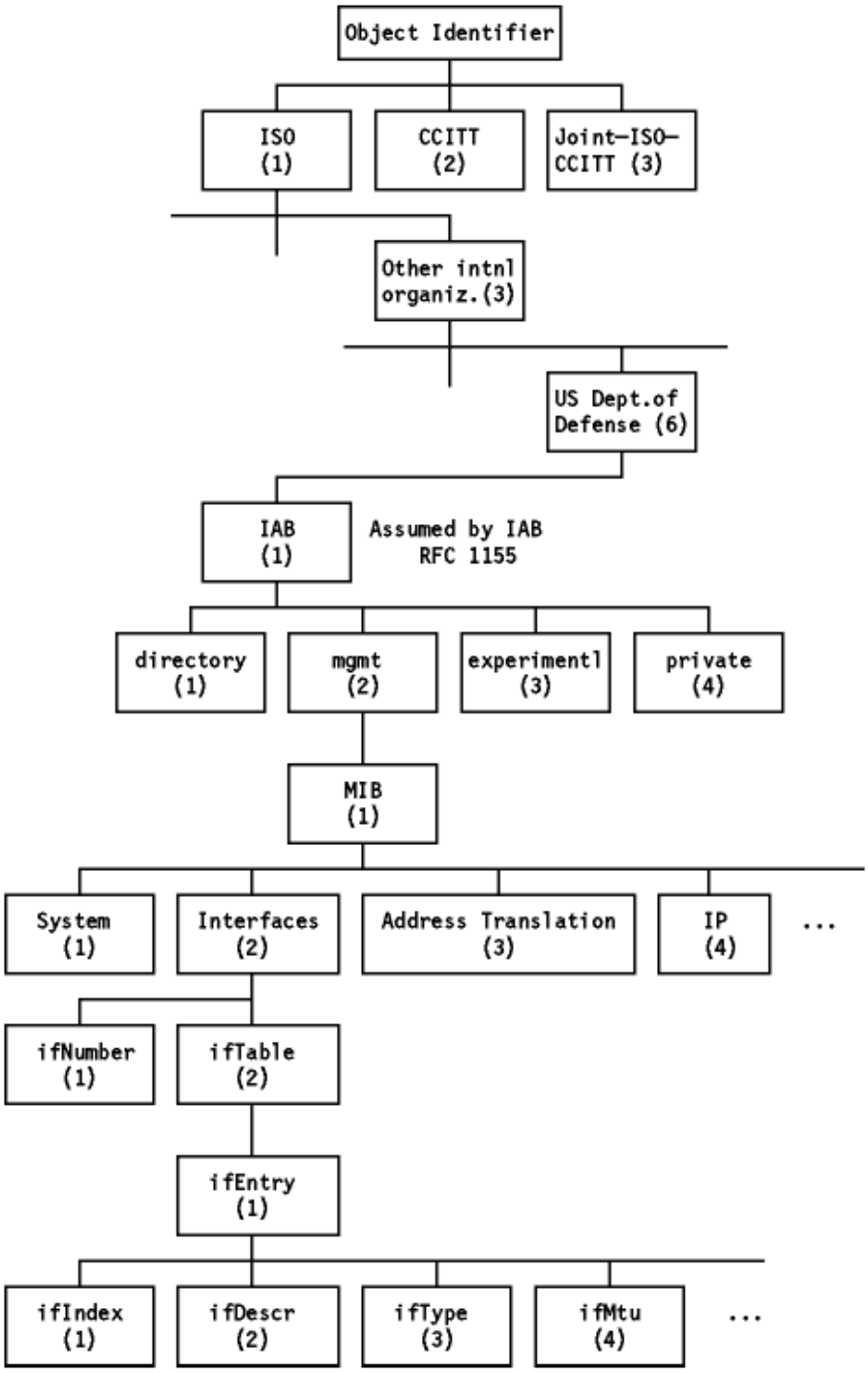
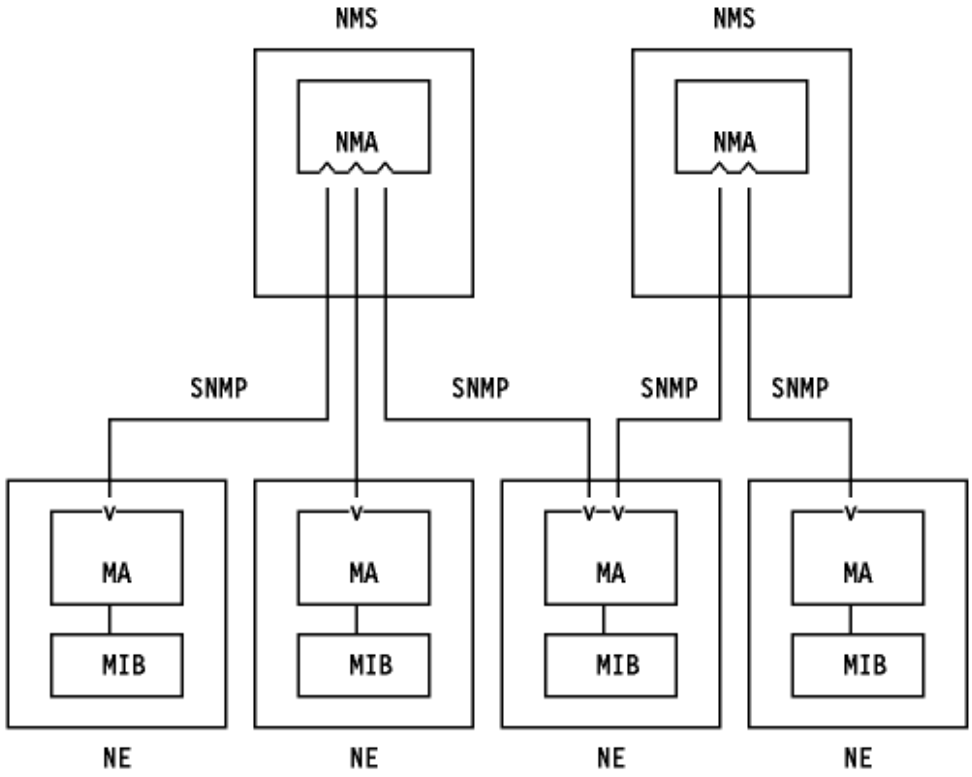


Figura: Identificador de objeto - Asignación para redes TCP/IP.

4.14.4 SNMP("Simple Network Management Protocol")

SNMP añadió las mejoras de muchos años de experiencia con SGMP y le permitió trabajar con los objetos definidos en el MIB con la representación del SIM.

ElRFC 1157 define NMS("Network Management Station") como una estación que ejecuta aplicaciones de gestión de red(NMA) que monitorizan y controlan elementos de red(NE) como hosts, pasarelas y servidores de terminales. Estos elementos usan un agente de gestión(MA) para realizar estas funciones. El SNMP para la comunicación de información entre las NMS y los MA.



NMS – Network Management Station
NMA – Network Management Application
NE – Network Element
MA – Management Agent
MIB – Management Information Base

Figura: SNMP - Componentes de SNMP.

Todos las funciones de los MA son sólo alteraciones(set) o consultas(get) de variables, limitando así el número de funciones esenciales a dos y simplificando el protocolo. En la comunicación NE-NMS, se utilizan un número limitado de mensajes no solicitados(traps) para informar de eventos asíncronos. Del mismo modo, en un intento de mantener la sencillez, el intercambio de información requiere sólo un servicio de datagramas y cada mensaje se envía en un único datagrama. Esto significa que SNMP es adecuado para una gran variedad de protocolos de transporte. El RFC 1157 especifica el intercambio de mensajes vía UDP, aunque es posible emplear otros.

Las entidades que residen en las NMS y los elementos de red que se comunican con otros a través de SNMP se denominan entidades de aplicación de SNMP. Los procesos que las implementan son las entidades de protocolo. Un agente SNMP con un conjunto arbitrario de entidades es una comunicad SNMP, en la que cada entidad se nombra con una ristra de bytes que debe ser unívoca para esa comunidad.

Un mensaje de SNMP consiste en un identificador de la versión, un nombre de la comunidad SNMP y un PDU("protocol data unit"). Toda implementación de SNMP debe soportar las cinco PDUs siguientes:

- **GetRequest**: Recuperar los valores de un objeto del MIB
- **GetNextRequest**: Recorrer parte del MIB
- **SetRequest**: Alterar los valores de un objeto del MIB
- **GetResponse**: Respuesta de GetRequest, GetNextRequest y SetRequest
- **Trap**: Capacidad de los elementos de red para generar eventos como la inicialización., reinicio o fallo en el enlace del MA. Hay siete tipos de traps definidos en el RFC 1157: coldStart, warmStart, linkDown, linkUp, authenticationFailure, egpNeighborLoss y enterpriseSpecific.

Los formatos de estos mensajes son los siguientes:

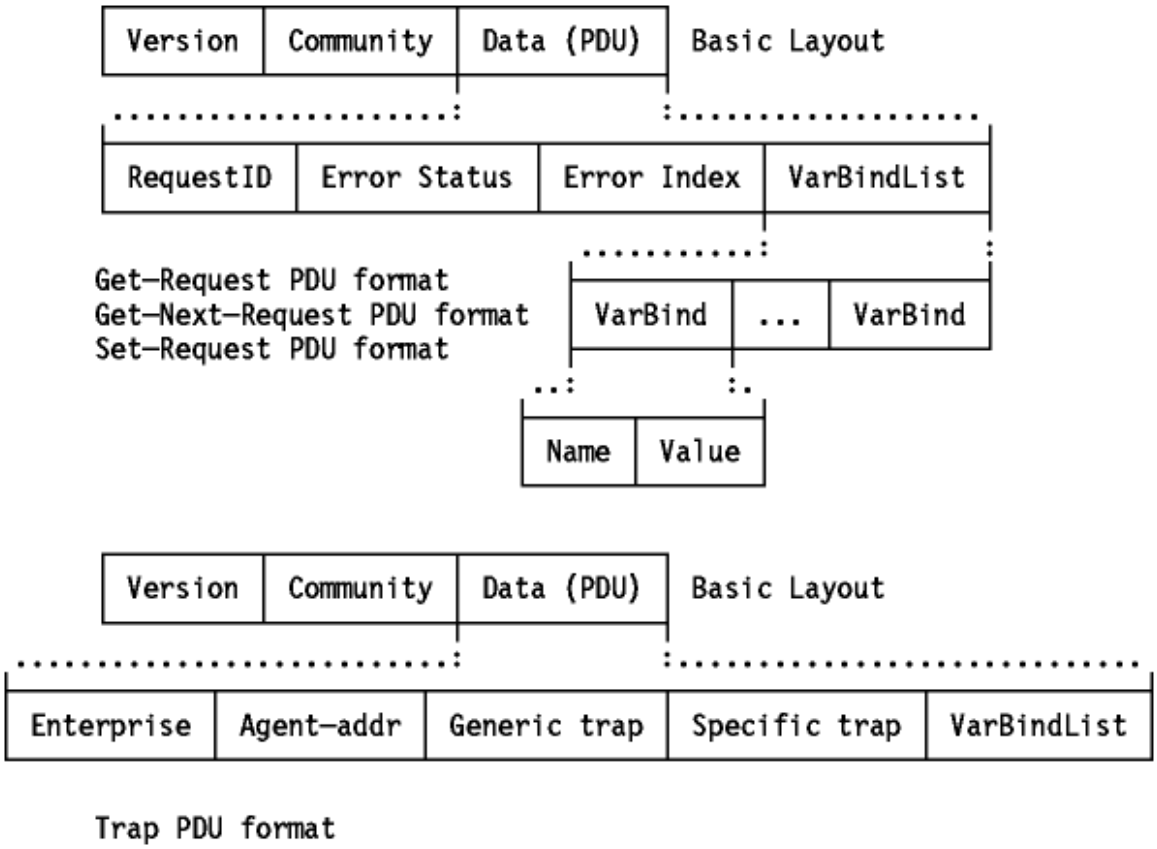


Figura: Formato de mensaje SNMP - Formato de las PDU Request, Set y Trap.

4.14.5 CMOT("Common Management Information Protocol over TCP/IP")

CMOT es la arquitectura de gestión de red desarrollada con vistas a mantener una relación más estrecha con el CMIP("Common Management Information Protocol") de OSI("Open System Interconnection"). Con esta premisa, CMOT se divide, como en OSI, en un modelo organizacional, funcional e informacional.

En los dos primeros el mismo concepto de OSI se usa en CMOT y SNMP. La identificación de objetos se efectúa empleando el subárbol relacionado con DoD con subdivisiones en lo que respecta a gestión, directorio, experimental y privado. Todos los objetos de gestión se definen en el MIB("Management Information Base"), y se representan con el SMI("Structure and Identification of Management Information"), un subconjunto de ASN.1("Abstract Syntax Notation 1" de OSI).

En el modelo funcional, CMOT adopta el modelo OSI que divide los componentes de gestión en managers y agentes. El agente recoge información, realiza comandos y ejecuta tests, y el manager recibe datos, genera comandos y envía instrucciones a los agentes. El manager y el agente están constituidos por un conjunto específico de entidades de información de gestión por cada capa de comunicación, denominadasLME ("Layer Management Entities").

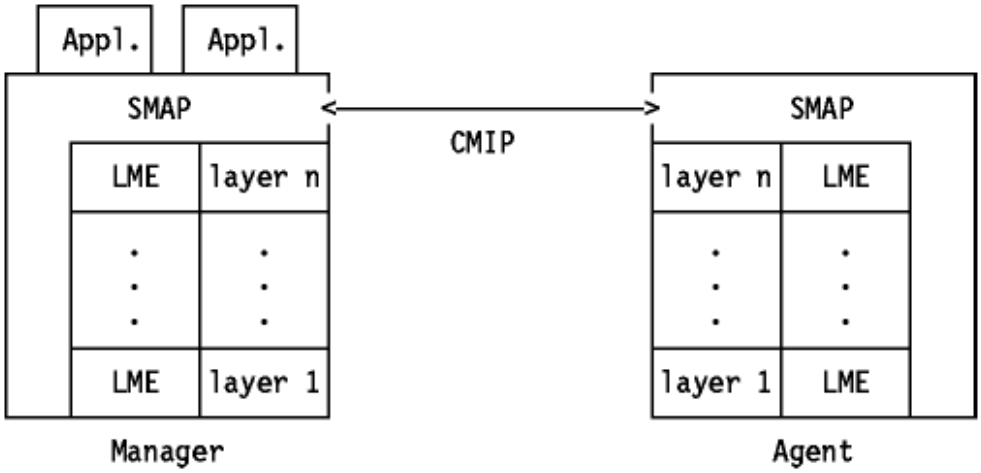


Figura: CMOT - Componentes de CMIP sobre TCP/IP.

Todos los LME los coordina el SMAP("System Management Application Process") que es capaz de comunicarse entre diferentes sistemas a través de CMIP("Common Management Information Protocol").

En el mundo OSI, la gestión sólo se puede producir sobre conexiones establecidas por completo entre managers y agentes. CMOT permite el intercambio de información de gestión usando servicios no orientados a conexión(datagramas). Pero para mantener la misma interfaz del servicio que requiere CMIP, llamada CMIS("Common Management Information Services"), la arquitectura de CMOT define una nueva capa, el LPP("Lightweight Presentation Protocol"). Esta capa se ha definido para proporcionar los servicios de presentación que necesita CMIP de tal forma que la totalidad de los estándares OSI para la gestión de red se adapten a la arquitectura TCP/IP de CMOT.

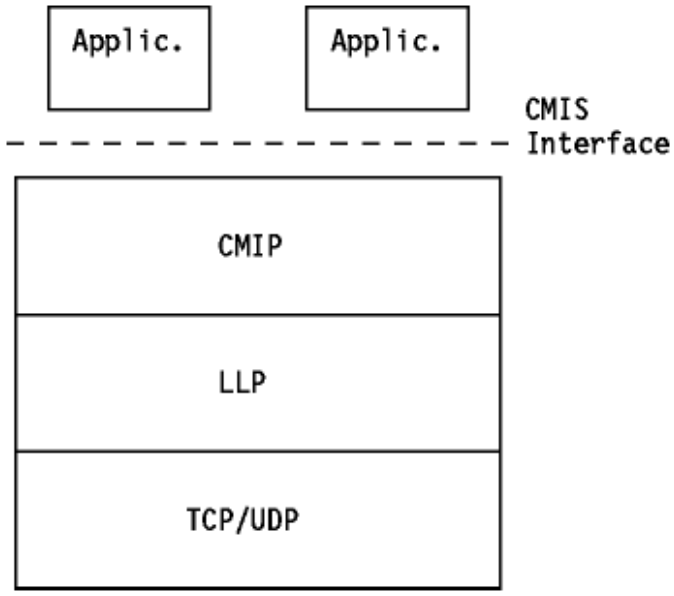


Figura: LPP("Lightweight Presentation Protocol")

4.14.6 El DPI de SNMP("SNMP Distributed Programming Interface")

SNMP define un protocolo que permite efectuar operaciones en una serie de variables. Este conjunto de variables(el MIB) y un conjunto básico o núcleo está predefinidas. Sin embargo, el diseño del MIB cuenta con la posibilidad de expandir este núcleo sea expandido. Desafortunadamente, las implementaciones convencionales de agentes SNMP no suministran mecanismos para que el usuario cree nuevas variables. El DPI enfoca esta cuestión proporcionando mecanismos que permiten al usuario añadir, borrar o reemplazar dinámicamente variables en el MIB local sin tener que recompilar el agente SNMP. Esto es posible gracias a un subagente que se comunica con el agente a través del DPI. El RFC 1228 lo describe.

El DPI de SNMP habilita a un proceso para registrar la existencia de una variable MIB en el agente SNMP, que pasará la solicitud al subagente. El subagente devuelve a su vez la respuesta apropiada al agente. Este, finalmente, empaqueta una respuesta SNMP y envía la respuesta a la NMS que inició la solicitud. El subagente es completamente invisible (transparente) para la NMS.

La comunicación entre el agente SNMP y sus clientes(subagentes) tiene lugar sobre un canal. Típicamente se trata de una conexión TCP, pero se pueden emplear otros protocolo de transporte orientados a conexión.

El agente en el DPI puede:

- Crear y borrar subárboles del MIB
- Crear un paquete de solicitud de registro para que el subagente informe al agente SNMp
- Crear un paquete de respuesta para que el subagente responda a la solicitud del agente SNMP
- Crear un paquete de solicitud TRAP

La siguiente figura muestra el flujo entre el agente SNMP y el subagente.

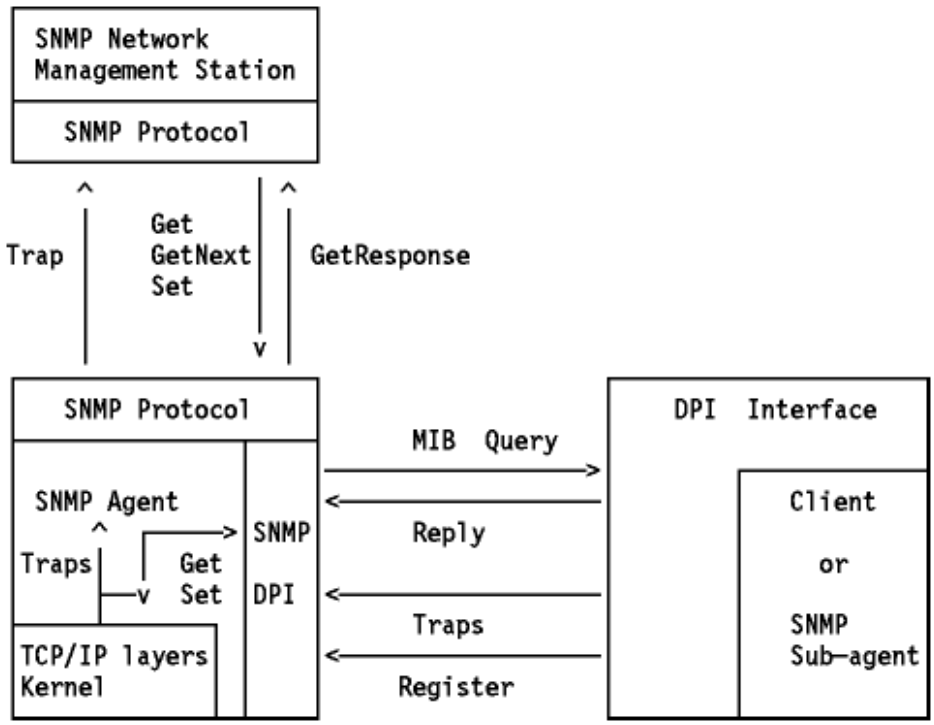


Figura: Descripción del DPI de SNMP

- El agente SNMP se comunica con el manager por medio de SNMP.
- La comunicación del agente con las capas TCP/IP y con el núcleo del sistema operativo depende de la implementación.

- Un subagente SNMP, ejecutando un proceso aparte(que potencialmente puede estar en otra máquina), puede registrar objetos con el agente SNMP(Register).
- El agente SNMP decodificará los paquetes. Si un paquete contiene una solicitud Get/GetNext o Set para un objeto registrado en el subagente, se la enviará en el correspondiente paquete(MIB query).
- El subagente SNMP responde con un paquete RESPONSE(Reply).
- El agente codifica la respuesta en un paquete SNMP y lo envía al manager.
- Si el subagente desea informar de un cambio de estado importante, envía un Trap al agente que a su vez lo codificará y enviará al manager.

4.14.7 SNMPv2("Simple Network Management Protocol, Versión 2")

La infraestructura de la versión 2 de SNMP se publicó en abril de 1993 y consiste en 12 RFCs, incluyendo el primero, el 1441, que es una introducción. En agosto de 1993 los 12 RFCs se convirtieron en un estándar con status electivo.

Esta infraestructura consta de las siguientes disciplinas:

- SMI("Structure of Management Information")

  Definición del subconjunto de ASNN.1 para la creación de módulos MIB. Descripción en el RFC 1442.

- Convenios textuales

  Definición del conjunto inicial de convenios textuales disponible para todos los módulos MIB. Descripción en el RFC 1443.

- Operaciones del protocolo

  Definición de las operaciones del protocolo con respecto a las PDUs enviadas y recibidas en el RFC 1448.

- Mapeados de transporte

  Definición del mapeado de SNMPv2 sobre un conjunto inicial de dominios de transporte ya que se puede utilizar en diferentes pilas de protocolo. El mapeado en UDP es el preferido. El RFC también define OSI, AppleTalk, IPX, etc. Descripción en el RFC 1449.

- Instrumentación del protocolo

  Definición del MIB y del MIB Manager-Manger. Descripción en los RFCs 1450 y 1451.

- Infraestructura administrativa

  Definición de SNMPv2 Party, SP("Security Protocols") y Party MIB. Descripción en los RFCs 1445, 1446 y 1447.

- Compatibilidades

  Definición de la *compatibilidad* o *capacidad* de notación de los agentes. Descripción en el RFC 1444.

Las siguientes secciones describen las principales diferencias y mejoras desde SNMPv1 a SNMPv2.

4.14.7.1 Entidad SNMPv2

Una entidad SNMPv2 es un proceso real que realiza operaciones de gestión de red mediante la generación y/o respuesta a/de mensajes SNMPv2. Todas las posibles operaciones de una entidad se pueden restringir a un subconjunto de las operaciones que puede efectuar el entorno de gestión("SNMPv2 Party" o EG). Remitirse aSNMPv2 Party más abajo. Una entidad SNMPv2 podría pertenecer a múltiples entidades gestoras, y mantiene las siguientes bases de datos locales:

- Una base de datos para todos los EG que conoce la entidad, que podrían ser:
  - Operación local
  - Operación local realizada por interacciones con EG o dispositivos remotos
  - Operación realizada por otras entidades SNMPv2
- Otra base de datos que representa todos los recursos de los objetos gestionados que conoce la entidad
- Como mínimo, una base de datos que representa una política de control de acceso que define los privilegios de acceso de acuerdo con los EG conocidos

Una entidad SNMPv2 puede actuar como agente o como manager SNMPv2.

4.14.7.2 Entorno de gestión("SNMPv2 Party" o EG)

Un entorno de gestión es un entorno de ejecución virtual cuyas operaciones se restringen, por razones de seguridad o de otra índole, a un subconjunto definido administrativamente de todas las operaciones que puede realizar una entidad SNMPv2 particular. Remitirse a Entidad SNMPv2 más arriba. Arquitectónicamente, cada EG comprende:

- Una identidad unívoca del entorno
- Una localización lógica de red en la que se ejecuta el EG, caracterizada por un dominio del protocolo de transporte y por información de direccionamiento del nivel de transporte
- Un sólo protocolo de autentificación y parámetros asociados con los que se autentifican el origen y la integridad de los mensajes del protocolo generados por el entorno
- Un sólo protocolo de privacidad y parámetros asociados con los que los mensajes de protocolo que recibe el entorno se protegen de cualquier intrusión

4.14.7.3 GetBulkRequest

El GetBulkRequest está definido en el RFC 1448 y forma por tanto parte de las operaciones del protocolo. Un mensaje GetBulkRequest se genera y se transmite como una petición de una aplicación SNMPv2. Su fin es solicitar la transferencia de una cantidad de datos potencialmente elevada, incluyendo, sin que ello le condicione, la rapidez y eficiencia en la recuperación de grandes tablas. GetBulkRequest es más eficiente que GetNextRequest en la recuperación de grandes tablas MIB de objetos. Su sintaxis es:

```
GetBulkRequest [ non-repeaters = N, max-repetitions = M ]
                ( RequestedObjectName1,
```

```
                RequestedObjectName2,
                RequestedObjectName3 )
```

Where:

RequestedObjectName1, 2, 3
: Identificador MIB del objeto, como sysUpTime, etc. Los objetos están en una lista ordenada léxicamente. Cada identificador de objeto está ligado como mínimo a una variable. Por ejemplo, el identificador *ipNetToMediaPhysAddress* está ligado a una variable para cada dirección IP de la tabla ARP y su contenido es la dirección MAC asociada.

N
: Especifica el valor de non-repeaters, lo que significa que se solicita sólo el contenido de la variable inmediata al objeto indicado en la solicitud, para los primeros N objetos nombrados entre paréntesis. Se trata de la misma función que desempeña GetNextRequest.

M
: Especifica el valor max-repetitions, lo que significa que se solicita del resto de los objetos(habiéndose solicitado N) el contenido de las M variables inmediatas al objeto indicado en la solicitud. Es similar a un GetNextRequest iterado pero transmitido en una sola solicitud.

Con GetBulkRequest se pueden conseguir los valores de sólo la siguiente variable o de las siguientes M variables con una sola solicitud.

Asumiendo la siguiente tabla ARP en un host que ejecuta un agente NMPv2:

```
Interface-Number  Network-Address  Physical-Address  Type

       1            10.0.0.51     00:00:10:01:23:45  static
       1             9.2.3.4      00:00:10:54:32:10  dynamic
       2            10.0.0.15     00:00:10:98:76:54  dynamic
```

Un manager SNMPv2 envía la siguiente respuesta para conseguir sysUpTime y la tabla ARP completa:

```
GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
               ( sysUpTime,
                 ipNetToMediaPhysAddress,
                 ipNetToMediaType )
```

La entidad SNMPv2 que actúa como agente responde con la PDU Response:

```
Response (( sysUpTime.0 =  "123456" ),
          ( ipNetToMediaPhysAddress.1.9.2.3.4 =
                                "000010543210" ),
          ( ipNetToMediaType.1.9.2.3.4 =  "dynamic" ),
          ( ipNetToMediaPhysAddress.1.10.0.0.51 =
                                 "000010012345" ),
          ( ipNetToMediaType.1.10.0.0.51 =  "static" ))
```

La entidad SNMPv2 que hace de manager continúa con:

```
GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
               ( sysUpTime,
                 ipNetToMediaPhysAddress.1.10.0.0.51,
                 ipNetToMediaType.1.10.0.0.51 )
```

El agente responde con:

```
Response (( sysUpTime.0 =  "123466" ),
          ( ipNetToMediaPhysAddress.2.10.0.0.15 =
                               "000010987654" ),
          ( ipNetToMediaType.2.10.0.0.15 =
                                    "dynamic" ),
          ( ipNetToMediaNetAddress.1.9.2.3.4 =
                                    "9.2.3.4" ),
          ( ipRoutingDiscards.0 =  "2" ))
```

Esta respuesta señala el final de la tabla al manager. Con GetNextRequest se hubieran necesitado cuatro solicitudes para conseguir la misma información. Si se hubiera fijado el valor *max-repetition* de GetBulkRequest a tres, en este ejemplo sólo se hubiera necesitado una solicitud.

4.14.7.4 InformRequest

Un mensaje InformRequest se genera y se transmite como una solicitud de una aplicación de una entidad manager SNMPv2 que desea notificar a otra aplicación, que se ejecuta también en un manager SNMPv2, información en el ámbito del MIB(MIB view) (20) para un entorno local a la aplicación que envía el mensaje. El paquete se utiliza para indicar al manager del otro entorno de la información accesible en el emisor. (comunicación manager-manager a través de los límites del entorno). Las dos primeras variables en la lista de asociaciones de variables de un mensaje InformRequest son sysUpTime.0 y snmpEventID.i (21)respectivamente. Les pueden seguir otras variables.

4.14.8 El MIB para SNMPv2

Este MIB define los objetos gestionados que determinan el comportamiento de la entidadSNMPv2.

Nota: No es una sustitución del MIB-II.

Las siguientes son algunas definiciones de objetos para hacerse una idea de sus contenidos:

```
snmpORLastChange OBJECT-TYPE
    SYNTAX     TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "El valor de sysUpTime en el momento
           del cambio más reciente en el valor
           o estado de cualquier instancia de
           snmpORID."

warmStart NOTIFICATION-TYPE
    STATUS  current
    DESCRIPTION
            "Un trap warmStart significa que la entidad
           SNMPv2, actuando como agente, se está reinicializando
           a sí misma de tal modo que la configuración
           no se altere."
```

4.14.9 EG del MIB("Party MIB")

El EG del MIB define los objetos gestionados que se corresponden con las propiedades asociadas a un EG SNMPv2. Un ejemplo de algunos objetos del MIB:

```
partyIdentity OBJECT-TYPE
    SYNTAX      Party
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Un identificador de EG unívoco
           para un EG de SNMPv2 particular."

partyAuthProtocol OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "El protocolo de autentificación
            por el que se autentifican el
            origen y la integridad de todos
            los mensajes que genera el EG. El
            valor noAuth significa que los
            mensajes no están autentificados.
            Una vez que se crea una instancia
            de este objeto, su valor no puede
            ser alterado."
```

4.14.9.1 MIB Manager-Manager

La finalidad de este MIB es proporcionar los medios para la coordinación entre múltiples estaciones de gestión. Es decir, los medios por los que las funciones de control y monitorización de la gestión de red se pueden distribuir entre múltiples NMS en una gran red. Específicamente, este MIB suministra mecanismos para que una NMS solicite servicios de gestión de otra. Por tanto, una entidad SNMPv2 puede tener un doble papel; cuando proporciona información de gestión a otro manager, actúa como agente, y cuando pide información, actúa como manager. El MIB manager-manager consta de las tres tablas siguientes:

- Alarmas
- Eventos
- Notificaciones

Cada alarma es una condición específica detectada mediante la monitorización periódica, en un intervalo de muestreo configurable, de los valores de una determinada variable con información de gestión. Un ejemplo de condición de alarma es cuando la variable monitorizada toma un valor fuera de rango. Cada condición de alarma dispara un evento, que puede a su vez desencadenar una o más notificaciones para otras NMS usando el InformRequest.

4.14.10 SAPP("Single Authentication and Privacy Protocol")

El protocolo de autentificación proporciona un mecanismo para que la gestión de SNMPv2 permita identificar que las comunicaciones que genera un entorno se originan efectivamente en ese entorno.

El protocolo de autentificación proporciona un mecanismo para que la gestión de SNMPv2 permita proteger las comunicaciones que genera un entorno de cualquier intrusión.

Las principales amenazas contra las que el protocolo de seguridad de SNMPv2 aporta protección son:

- Modificación de información
- Enmascaramiento
- Modificación del flujo de mensajes
- Intrusión en la información

Los siguientes servicios de seguridad proporcionan medidas contra las anteriores amenazas:

- Integridad de los datos

  La proporciona el algoritmo de condensación de mensajes MD5. Se calcula un resumen o extracto de 128 bits de la porción indicada del mensaje SNMPv2 y se incluye como parte del mensaje enviado al receptor.

- Autentificación del origen de los datos

  A cada mensaje se le añade un prefijo con un valor secreto que comparten el emisor del mensaje y el receptor, antes de calcular el extracto.

- Replay o retardo del mensaje

  En cada mensaje se incluye un sello de tiempo,

- Confidencialidadd de los datos

  La proporciona el protocolo simétrico de privacidad que encripta una porción adecuada del mensaje de acuerdo con una llave secreta conocida sólo por el emisor y el receptor. Este protocolo se usa conjuntamente con el algoritmo simétrico de encriptación, en el modo de encadenamiento de cifrado de bloques, que forma parte del DES("Data Encryption Standard"). La parte designada del mensaje se encripta y se incluye como parte el mensaje enviado el receptor.

4.14.11 El nuevo modelo administrativo

Uno de los propósitos del modelo administrativo para SNMPv2 es definir como la infraestructura administrativa se aplica para llevar a cabo una administración de red efectiva en diversas configuraciones y entornos.

El modelo implica el uso de diferentes identidades en el intercambio de mensajes. De esta forma, representa abandonar el basado en comunidades del SNMPv1 original. Al identificar sin ambigüedad al emisor y al receptor de cada mensaje, esta nueva estrategia mejora el esquema histórico de comunidades ya que permite un diseño del control de acceso a los datos más conveniente así como el empleo de protocolos de seguridad asimétricos(con llave pública) en el futuro. Remitirse a Figura - Formato de mensaje de SNMPv2 para conocer el nuevo formato de mensaje.

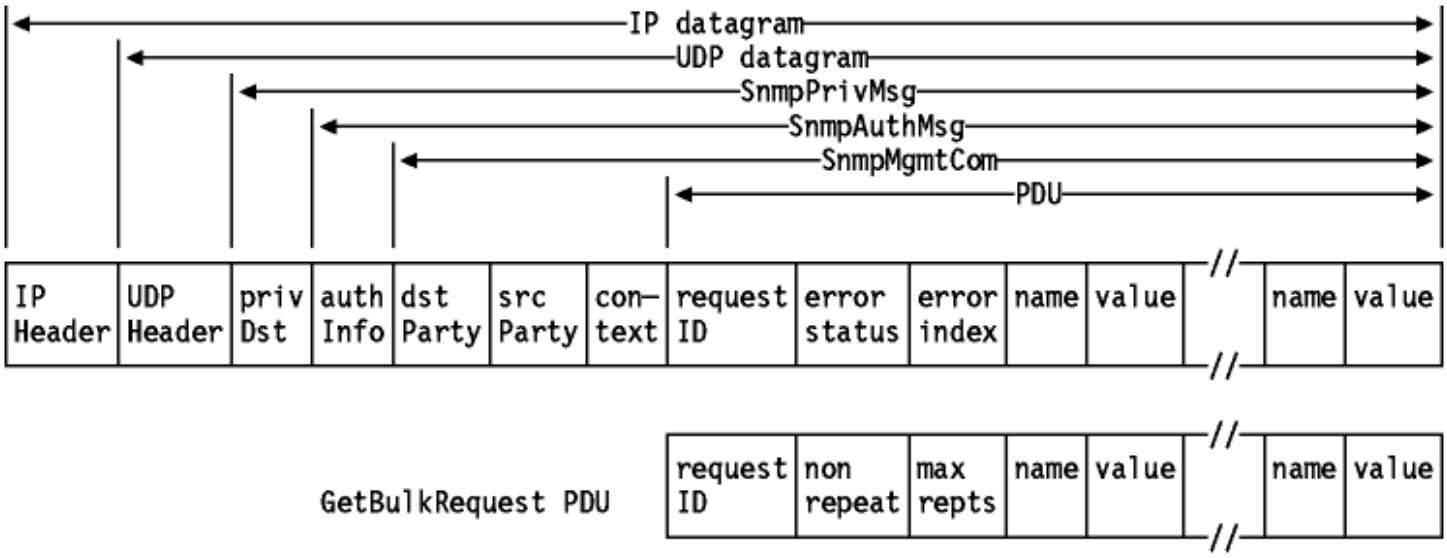


Figura: Formato de mensaje de SNMPv2

PDU
: Incluye una de las siguientes PDUs
  - GetRequest
  - GetNextRequest
  - Response
  - SetRequest
  - InformRequest
  - SNMPv2-Trap

  El GetBulkRequest tiene un formato de PDU distinto al mostrado más arriba. Ver GetBulkRequest.

  Nota: El SNMP-Trap tiene ahora el mismo formato que las demás solicitudes.

SnmpMgmtCom ("SNMP Management Communication")
: Añade el identificador del entorno emisor(srcParty), del receptor(dstParty) y el contexto a la PDU. El contexto especifica el ámbito de SNMPv2 que contiene la información de gestión a la que referencia la comunicación.

SnmpAuthMsg
: Este campo se utiliza como información de autentificación para el protocolo de información usado por el entorno en cuestión. El SnmpAuthMsg está serializado de acuerdo con ASN.1 BER (22) por lo que puede ser encriptado.

SnmpPrivMsg SNMP Private Message
: El SNMPv2 Private Message es un mensaje SNMPv2 autentificado que posiblemente está protegido de intrusiones en la información que contiene. Un destino privado(privDst) se añade al entorno de destino.

El mensaje pasa a ser encapsulado en un datagrama UDP/IP normal y se envía a su destino a través de la red.

Para más información, remitirse a los RFCs mencionados arriba.

Tabla de contenidos Protocolo de servicios NetBIOS

 Tabla de contenidos  El nuevo modelo administrativo

4.15 Protocolo de servicios NetBIOS

El protocolo de servicios es un *protocolo estándar*. Su status es *electivo*.

Los RFCs 1001 y 1002 describen la implementación estándar de los servicios NetBIOS de IBM(como se explica en *"IBM Technical Reference PC Network"*) sobre los protocolos TCP y UDP. Recordar que la mayoría de las aplicaciones actuales para redes IBM en anillo utilizan los servicios NetBIOS para comunicarse en estas redes. Ejemplos son el "OS/2 LAN Server" y el "DOS LAN Requester". Entre las aplicaciones OEM que emplean los servicios NetBIOS está el Lotus Notes.

La mayoría de las aplicaciones NetBIOS de hoy en día corren sobre PCs y PS/2s. La implementación de NetBIOS en TCP/IP proporcionaría la potencia de los ordenadores grandes y pequeños a los usuarios de los PC. Una aplicación típica sería el uso de grandes sistemas como servidores de ficheros.

 Tabla de contenidos  LPD ("Line Printer Daemon")

Tabla de contenidos Protocolo de servicios NetBIOS

4.16 LPD("Line Printer Daemon")

El protocolo LPD("Line Printer Daemon protocol") es un *protocolo informacional*. Su status es *de uso limitado*. La especificación actual se halla en el RFC 1179. Notar que este RFC no especifica un estándar de Internet. Sin embargo, esta función ha formado parte de los sistemas UNIX durante mucho tiempo(en *uucp*).

Tabla de contenidos Protocolo BOOTstrap - BOOTP

4.17 Protocolo BOOTstrap - BOOTP

BOOTP es un *borrador de protocolo de Internet*. Su status es *recomendado*. Las especificaciones de BOOTP se pueden encontrar en los *RFC 951 - Bootstrap* y *RFC 1497 - Extensiones de la información de los distribuidores de BOOTP*.

Además hay actualizaciones de BOOTP que lo permiten interoperar con DHCP(ver DHCP("Dynamic Host Configuration Protocol")).Estas se describen en el *RFC 1542 - Aclaraciones y extensiones para el protocolo Bootstrap* que actualiza a los RFC 951 y *RFC 1533 - Opciones y extensiones de DHCPs*, que desfasa al RFC 1497. Dichas actualizaciones de BOOTP son *estándares propuestos* con status *electivo*.

Las LANs hacen posible usar host sin disco como estaciones de trabajo, "routers" concentradores de terminales, etc. Los host sin disco requieren de algún mecanismo para el arranque remoto sobre una red. El protocolo BOOTP se utiliza para efectuar arranques remotos en redes IP. Permite que una pila de IP mínima sin información de configuración, típicamente almacenada en la ROM, obtenga información suficiente para comenzar el proceso de descargar el código de arranque necesario. BOOTP no define como se realiza esta descarga, pero habitualmente se emplea TFTP (ver también TFTP("Trivial File Transfer Protocol")) como se describe en el *RFC 906 - Carga en Bootstrap usando TFTP*.

El proceso BOOTP implica los siguientes pasos:

1. El cliente determina su propia dirección hardware; esta suele estar en una ROM del hardware.
2. El cliente BOOTP envía su dirección hardware en un datagrama UDP al servidor. El contenido completo de este datagrama se muestra en Figura - Formato de mensaje de BOOTP. Si el cliente conoce su dirección IP y/o la dirección del servidor, debería usarlas, pero en general los clientes BOOTP carecen de configuración IP en absoluto. Si el cliente desconoce su dirección IP, emplea la 0.0.0.0. Si desconoce la dirección IP del servidor, utiliza la dirección de broadcast limitado(255.255.255.255). El número del puerto UDP es el 67.
3. El servidor recibe el datagrama y busca la dirección hardware del cliente en su fichero de configuración, que contiene la dirección IP del cliente. El servidor rellena los campos restantes del datagrama UDP y se lo devuelve al cliente usando el puerto 68. Hay tres métodos posibles para hacer esto:
    - ❍ Si el cliente conoce su propia dirección IP(incluida en la solicitud BOOTP), entonces el servidor devuelve directamente el datagrama a esa dirección. Es probable que la caché de ARP en la pila de protocolos del servidor desconozca la dirección hardware correspondiente a esa dirección IP. Se hará uso de ARP para determinarla del modo habitual.
    - ❍ Si el cliente desconoce su propia dirección IP(0.0.0.0 la solicitud BOOTP), entonces el servidor se ocupa de averiguarla con su propia caché de ARP. El servidor no puede usar ARP para resolver la dirección hardware del cliente porque el cliente no sabe su dirección IP y por lo tanto no puede responder a una petición ARP. Es el problema de la pescadilla que se muerde la cola. Hay dos soluciones posibles:
        - ❑ Si el servidor tiene un mecanismo para actualizar directamente su propia caché ARP sin usar ARP, lo utiliza y envía directamente el datagrama.
        - ❑ Si el servidor no puede actualizar su propia caché, debe enviar una respuesta en forma de broadcast.
1. Cuando reciba la respuesta, el cliente BOOTP grabará su dirección IP(permitiéndole responder a peticiones ARP) y comenzará el proceso de arranque.

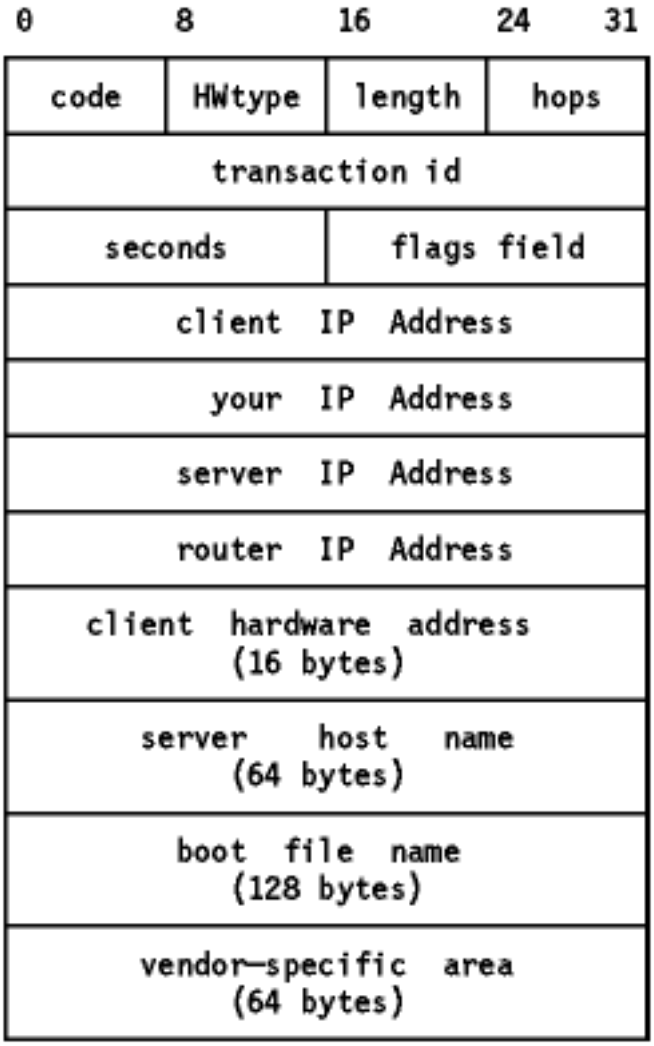


Figura: Formato de mensaje de BOOTP

code
: Indica una solicitud o una respuesta

1
: Request

2
: Reply

HWtype
: El tipo de hardware, por ejemplo:

1
: Ethernet

6
: IEEE 802 Networks

Remitirse a *STD 2 - Números asignados de Internet* para un alista completa.

length
: Longitud en bytes de la dirección hardware. Ethernet y las redes en anillo usan 6, por ejemplo.

hops
: El cliente lo pone a 0. Cada "router" que retransmite la solicitud a otro servidor lo incrementa, con el fin de detectar bucles. El RFC 951 sugiere que un valor de 3 indica un bucle.

Transaction ID
: Número aleatorio usado para comparar la solicitud con la respuesta que genera.

Seconds
: Fijado por el cliente. Es el tiempo transcurrido en segundos desde que el cliente inició el proceso de arranque.

Flags Field
: El bit más significante de este campo se usa como flag de broadcast. Todos los demás bits deben estar a 0; están reservados para usos futuros. Normalmente, los servidores BOOTP tratan de entregar los mensajes BOOTREPLY

directamente al cliente usando unicast. La dirección de destino en la cabecera IP se pone al valor de la *dirección IP* fijada por el servidor BOOTP, y la dirección MAC a la *dirección hardware* del cliente BOOTP. Si un host no puede recibir un datagrama IP en unicast hasta saber su propia dirección IP, el bit de broadcast se debe poner a 1 para indicar al servidor que el mensaje BOOTREPLY se debe enviar como un broadcast en IP y MAC. De otro modo, este bit debe ponerse a cero.

Client IP adress
: Fijada por el cliente. O bien es su dirección IP real , o 0.0.0.0.

Your IP Address
: Fijada por el servidor si el valor del campo anterior es 0.0.0.0

Server IP address
: Fijada por el servidor.

Router IP address
: Fijada por el "router" retransmisor si se usa retransmisión BOOTP.

Client hardware address
: Fijada por el cliente y usada por el servidor para identificar cuál de los clientes registrados está arrancando.

Server host name
: Nombre opcional del host servidor acabado en X'00'.

Boot file name
: El cliente deja este campo vacío o especifica un nombre genérico, tal como "router", indicando el tipo de archivo de arranque a usar. El servidor devuelve el nombre completo del fichero o bien el de un fichero de arranque adecuado para el cliente. Su valor tiene como terminador a la secuencia X'00.

Vendor-specific area
: Área específica del distribuidor(opcional). Se recomienda que el cliente llene siempre los cuatro primeros bytes con un *"magic cookie"* o galleta mágica. Si el cookie específica de un distribuidor no se usa, el cliente debería utilizar 99.130.83.99 seguido de una marca de fin(255) y fijar los bis restantes a cero. Remitirse al RFC 1533 para más detalles.

Una restricción a este esquema es el uso del broadcast limitado para las solicitudes BOOTP; requiere que el servidor esté en la misma subred que el cliente. La retransmisión BOOTP es un mecanismo para que los "routers" transmitan solicitudes BOOTP. Es una opción de configuración disponible en algunos "routers". Ver el RFC 951 para más información.

Una vez que el cliente BOOTP ha procesado la respuesta, puede proceder con la transferencia del fichero de arranque y ejecutar el proceso de arranque completo. Ver el RFC 906 para la especificación de como se hace esto con TFTP. El proceso de arranque completo reemplaza la pila mínima de IP usada por BOOTP y TFTP por una pila IP normal transferida como parte del fichero de arranque, que contiene la configuración correcta para el cliente.

 Tabla de contenidos  DHCP ("Dynamic Host Configuration Protocol")

4.18 DHCP("Dynamic Host Configuration Protocol")

DHCP es un *protocolo propuesto como estándar*. Su status es *electivo*. Las especificaciones actuales de DHCP se pueden encontrar en los *RFC 1541 - DHCP("Dynamic Host Configuration Protocol")* y *RFC 1533 - Opciones de DHCP y extensiones de los distribuidores de BOOTP*.

DHCP proporciona un entorno de trabajo para pasar información de configuración a hosts en una red TCP/IP. DHCP se basa en el protocolo BOOTP, añadiendo la capacidad de asignar automáticamente direcciones de red reutilizables y opciones de configuración adicionales. Para más información sobre BOOTP, remitirse a BOOTP(("BOOTstrap Protocol"). Los elementos participantes en DHCP pueden interoperar con los de BOOTP(RFC 1534).

DHCP consta de dos componentes:

1. Un protocolo que entrega parámetros de configuración específicos de un host de un servidor DHCP al host.
2. Un mecanismo para reservar direcciones de red para los hosts.

IP requiere la configuración de muchos parámetros dentro del software de implementación del protocolo. Debido a que IP utilizar en muchas clases distintas de hardware de red, no se puede suponer o adivinar que los valores de esos parámetros tienen valores correctos por defecto. El uso de un sistema de asignación de direcciones distribuidas basado en un mecanismo de consulta/defensa, para descubrir direcciones de red que ya están en uso, no garantiza direcciones de red unívocas porque puede que los host no sean siempre capaces de defender sus direcciones de red.

DHCP soporta tres mecanismos para la asignación de direcciones IP:

1. Asignación automática

   DHCP asigna al host una dirección IP permanente.

2. Asignación dinámica

   DHCP asigna una dirección IP por un periodo de tiempo limitado. Una red así se denomina *arrendamiento*. Este es el único mecanismo que permite la reutilización automática de direcciones que ya no son necesitadas por los hosts a los que estaban asignadas.

3. Asignación manual

   La dirección del host es asignada por el administrador de red

El formato del mensaje DHCP:

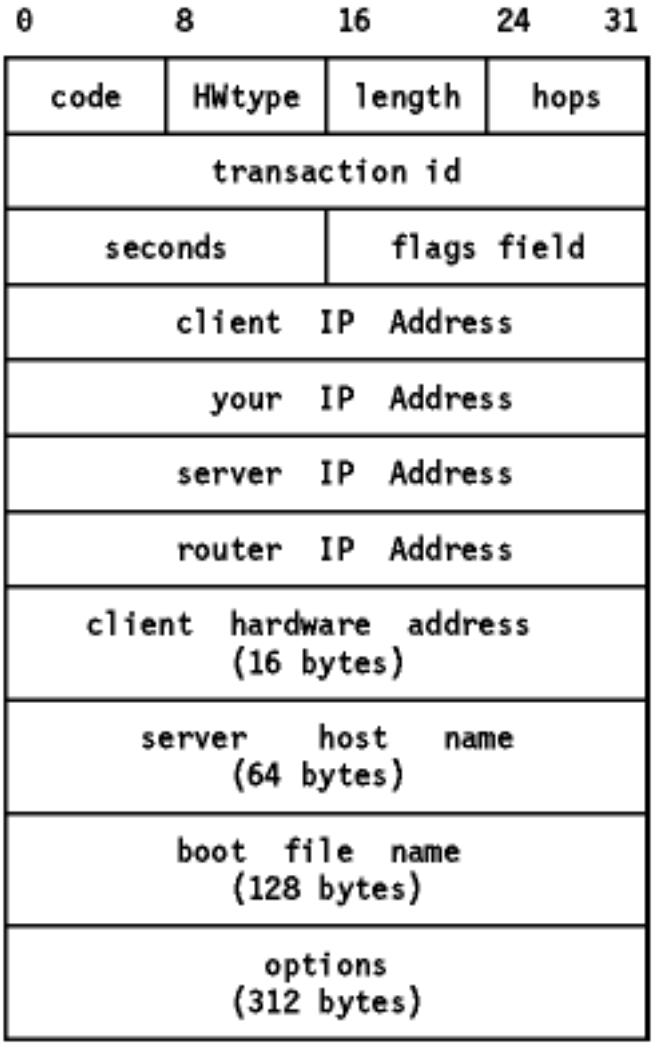


Figure: Formato del mensaje DHCP

code
: Indica solicitud o respuesta

1
: Request

2
: Reply

HWtype
: El tipo de hardware, por ejemplo:

1
: Ethernet

6
: IEEE 802 Networks

Remitirse al *STD 2 - Números asignados de Internet* para una lista completa.

length
: Longitud en bytes de la dirección hardware. Ethernet y las redes en anillo usan 6, por ejemplo.

hops
: El cliente lo pone a 0. Cada "router" que retransmite la solicitud a otro servidor lo incrementa, con el fin de detectar bucles. El RFC 951 sugiere que un valor de 3 indica un bucle.

Transaction ID
: Número aleatorio usado para comparar la solicitud con la respuesta que genera.

Seconds
: Fijado por el cliente. Es el tiempo transcurrido en segundos desde que el cliente inició el proceso de arranque.

Flags Field
: El bit más significante de este campo se usa como flag de broadcast. Todos los demás bits deben estar a 0; están reservados para usos futuros. Normalmente, los servidores DHCP tratan de entregar los mensajes DHCPREPLY directamente al cliente usando unicast. La dirección de destino en la cabecera IP se pone al valor de la *dirección IP* fijada

por el servidor DHCP, y la dirección MAC a la *dirección hardware* del cliente DHCP. Si un host no puede recibir un datagrama IP en unicast hasta saber su propia dirección IP, el bit de broadcast se debe poner a 1 para indicar al servidor que el mensaje DHCPREPLY se debe enviar como un broadcast en IP y MAC. De otro modo, este bit debe ponerse a cero.

Client IP adress
: Fijada por el cliente. O bien es su dirección IP real , o 0.0.0.0.

Your IP Address
: Fijada por el servidor si el valor del campo anterior es 0.0.0.0

Server IP address
: Fijada por el servidor.

Router IP address
: Fijada por el "router" retransmisor si se usa retransmisión BOOTP.

Client hardware address
: Fijada por el cliente y usada por el servidor para identificar cuál de los clientes registrados está arrancando.

Server host name
: Nombre opcional del host servidor acabado en X'00'.

Nombre del fichero de arranque
: El cliente o bien deja este campo vacío o especifica un nombre genérico, como "router" indicando el tipo de fichero de arranque a usar. En la solicitud de DHCPDISCOVER se pone al valor nulo. El servidor devuelve el la ruta de acceso completa del fichero en una respuesta DHCPOFFER. El valor termina en X'00'.

Options
: Los primeros cuatro bytes del campo de opciones del mensaje DHCP contienen el cookie(99.130.83.99). El resto del *campo* de opciones consiste en parámetros marcados llamados *opciones*. Remitirse al RFC 1533 para más detalles.

4.18.0.1 Almacén de parámetros de configuración

DHCP permite un almacenamiento persistente de los parámetros de red de los clientes. El modelo de almacenamiento persistente en DHCP consiste en que el servicio DHCP almacena una entrada con un valor y una clave para cada cliente, donde la clave es un identificador único, por ejemplo un número IP de subred y un identificador único dentro de la subred, y el valor contiene los parámetros de configuración del cliente.

4.18.0.2 Asignando un nueva dirección de red

Esta sección describe la interacción del cliente servidor cuando el cliente no conoce su dirección de red. Se asume que el servidor DHCP tiene un bloque de direcciones de red con el que puede satisfacer peticiones de nuevas direcciones. Cada servidor mantiene además una base de datos local y permanente de las direcciones asignadas y de los arrendamientos.

1. El cliente hace un broadcast de un mensaje DHCPDISCOVER en su subred física. El mensaje DHCPDISCOVER puede incluir algunas opciones como sugerencias de la dirección de red, duración del arrendamiento, etc.
2. Cada servidor puede responder con un mensaje DHCPOFFER que incluye una dirección de red disponible y otras opciones de configuración.
3. El cliente recibe uno o más mensaje DHCPOFFER de uno o más servidores. Elige uno basándose en los parámetros de configuración ofertados y hace un broadcast de un mensaje DHCPREQUEST que incluye la opción identificadora del servidor para indicar qué mensaje ha seleccionado.
4. Los servidores reciben el broadcast de DHCPREQUEST del cliente. Los servidores no seleccionados utilizan el mensaje como notificación e que el cliente ha declinado su oferta. El servidor seleccionado vincula al cliente al almacenamiento persistente y responde con un mensaje DHCPACK que contiene los parámetros de configuración para el cliente. La combinación de las direcciones hardware y asignada del cliente constituyen un identificador único de su arrendamiento y las usan tanto el cliente como el servidor para identificar cualquier arrendamiento al que se haga referencia en un mensaje DHCP. El campo "your IP address" en los mensaje DHCPACK se rellena con la dirección de red seleccionada.
5. El cliente recibe el mensaje DHCPACK con parámetro de configuración. Realiza un chequeo final de estos parámetros, por ejemplo con ARP para la dirección de red asignada, y registra la duración del arrendamiento y el cookie de identificación de éste especificado en el mensaje DHCPACK. En este punto, el cliente está configurado. Si el cliente detecta un problema con los parámetros en el mensaje DHCPACK, envía un mensaje DHCPDECLINE al servidor y reinicia el proceso de configuración. El cliente debería esperar un mínimo de diez segundos antes de reiniciar este proceso para evitar un exceso de tráfico en la red en caso de que se produzca algún bucle.

   Si el cliente recibe un mensaje, reinicia el proceso de configuración.

6. Puede elegir renunciar a su arrendamiento enviando un mensaje DHCPRELEASE al servidor. El cliente especifica el arrendamiento al que renuncia incluyendo sus direcciones hardware y de red

4.18.0.3 Reutilizando una dirección de red previamente asignada

Si el cliente recuerda y desea usar una dirección de red previamente asignada se llevan a cabo los siguientes pasos:

1. El cliente hace un broadcast de un mensaje DHCPREQUEST en su subred. Este mensaje incluye la dirección de red del cliente.
2. Los servidores que conozcan los parámetros de configuración del cliente le responden con un mensaje DHCPACK.
3. El cliente recibe el mensaje DHCPACK con parámetros de configuración. Efectúa un último chequeo de estos y registra la duración del arrendamiento y el cookie de identificación de este, especificado en el DHCPACK. En este punto, el cliente está configurado.

> Si el cliente detecta algún problema con los parámetros en el DHCPACK, envía un mensaje DHCPDECLINE al servidor y reinicia el proceso de configuración solicitando una nueva dirección de red. Si el cliente recibe un mensaje DHCPNAK, no puede reutilizar la dirección que solicitó. En vez de eso debe pedir una nueva dirección reiniciando el proceso de configuración descrito en Asignando una nueva dirección de red. El cliente puede elegir renunciar a su arrendamiento de la dirección de red al enviar un mensaje DHCPRELEASE al servidor. Identifica el arrendamiento al que renuncia con el cookie de identificación.

Nota: Un host debería usar DHCP para readquirir o verificar su dirección IP y sus parámetros de configuración siempre que cambien los parámetros de su red local, por ejemplo en el arranque del sistema o después de una desconexión de la red local, ya que la configuración de esta puede haber cambiado sin que lo sepa el host o el usuario.

Para más información, remitirse a los RFCs mencionados anteriormente.

Tabla de contenidos NETSTAT

Tabla de contenidos Reutilizando una dirección de red previamente asignada

4.19 NETSTAT

El comando NETSTAT se usa para consultar a TCP/IP acerca del estado de la red en la que se halla el host local. La sintaxis exacta de este comando depende mucho de la implementación. Ver la *Guía de usuario* o el *Manual de referencia de comandos* de la correspondiente implementación para más detalle. Se trata de un herramienta útil para la depuración.

En general, NETSTAT proporciona información sobre:

- Las conexiones TCP activas en el host local.
- El estado de todos los servidores TCP/IP del servidor local y de los zócalos que usan.
- Dispositivos y enlaces usados por TCP/IP.
- Las tablas de encaminamiento IP(tablas de las pasarelas) usadas en el host local.

Tabla de contenidos Protocolo Finger

Tabla de contenidos NETSTAT

4.20 Protocolo Finger

Finger es un *protocolo propuesto como borrador*. Su status es *electivo*. La especificación actual de Finger se encuentra en el *RFC 1288 - Información de usuario para el protocolo Finger*.

El comando finger muestra información sobre los usuarios de un host remoto. Finger es un comando de UNIX. Su formato es:
finger usuario@host
o
finger @host

La información que da el finger de un usuario depende de la implementación del servidor finger. Si no se especifica ningún usuario, la información será típicamente una lista de todos los usuarios conectados actualmente al host.

Las conexiones se establecen por el puerto 79(decimal) de TCP. El cliente envía un comando en forma de cadena de caracteres ASCII, acabada en <CRLF>. El servidor responde con una o más cadenas ASCII, hasta que cierra la conexión.

Tabla de contenidos Protocolo Whois

Tabla de contenidos Protocolo Finger

4.21 Protocolo Whois

Whois es un *protocolo propuesto como borrador*. Su status es *electivo*. La especificación actual de Whois se encuentra en el RFC *RFC 954 - NICNAME/WHOIS*.

El programa Whois se usa habitualmente en el entorno UNIX para conectarse a un servidor Whois. El propósito del servidor es proporcionar servicios de tipo directorio. El servidor Whois original se instaló de modo que el NIC("Network Information Center") pudiera mantener una lista de contactos para redes conectadas a Internet. Sin embargo, en la actualidad muchos sitios usan Whois para suministrar servicios de directorio locales.

El servidor Whois se basa en TCP y usa el puerto bien conocido número 43. Las solicitudes y las respuestas intercambiadas entre cliente y servidor utilizan el modo de terminal NVT ASCII.

InterNIC mantiene el Whois usado más ampliamente y se puede encontrar en rs.internic.net.

Tabla de contenidos Protocolos "time" y "daytime"

Tabla de contenidos Protocolo Whois

4.22 Protocolos "time" y "daytime"

El protocolo "time" es un *protocolo estándar*. Su status es *electivo*. La especificación actual de TIME se encuentra en el *RFC 868 - Protocolo del servidor "Time"*.

"Daytime" es un *protocolo estándar*. Su status es *electivo*. La especificación actual de DAYTIME se encuentra en el *RFC 867 - Protocolo "Daytime"*.

4.22.0.1 Concepto

El protocolo "time" proporciona la fecha y la hora en un formato legible para la máquina. Usa como capa de transporte TCP o UDP. Su puerto es el 37(decimal).

1. Si se usa vía TCP:
    - S: escuchar en el puerto 37
    - C: conectarse al puerto 37
    - S: enviar la hora como un número binario de 32 bits
    - C: recibir el número
    - C: cerrar la conexión
1. Si se usa vía UDP:
    - S: escuchar en el puerto 37
    - C: enviar un datagrama vacío al puerto 37
    - S: recibir el datagrama vacío
    - S: enviar el datagrama con la hora en forma de número binario de 32 bits
    - C: recibir el datagrama

El número de 32 bits representa el número de segundos transcurridos desde la medianoche del 1 de enero de 1900(GMT). Es conveniente para que la máquina sincronice su propio reloj, pero no resulta legible a ojos del usuario.

El protocolo "daytime" es similar en su concepción, pero el servidor enviará al cliente una cadena ASCII que al usuario le resultará legible. El formato de esta cadena no se halla formalizado. Para este protocolo se usa el puerto 13(decimal).

Tabla de contenidos Otros Protocolos de aplicación

4.23 Otros protocolos de aplicación

Hay una serie de otros protocolos de aplicación que no están documentados en los RFCs , pero que se implementan en diversos productos. Las siguientes secciones describen algunos de estos protocolos.

4.23.1 NDB("Network Database")

El protocolo NDB("Network Database") no es un estándar de Internet.

NDB define un protocolo para orientado a sistemas de bases de datos relacionales en entorno TCP/IP. Tiene los siguientes objetivos:

- Permitir a los usuarios y aplicaciones de cualquier estación de trabajo o mainframe lanzar sentencias SQL, bien de modo interactivo o embebidas en los programas de aplicación, para acceder cualquier base de datos en cualquier sistema operativo.
- Distribuir las aplicaciones de la base de datos de un host a una máquina concreta.

NDB se construye sobre RPC("Remote Procedure Call"), en cualquiera de sus implementaciones, utilizando el modelo cliente/servidor. Ver Figura - Componentes del protocolo NDB("Network Database") para una descripción de los diversos componentes de NDB

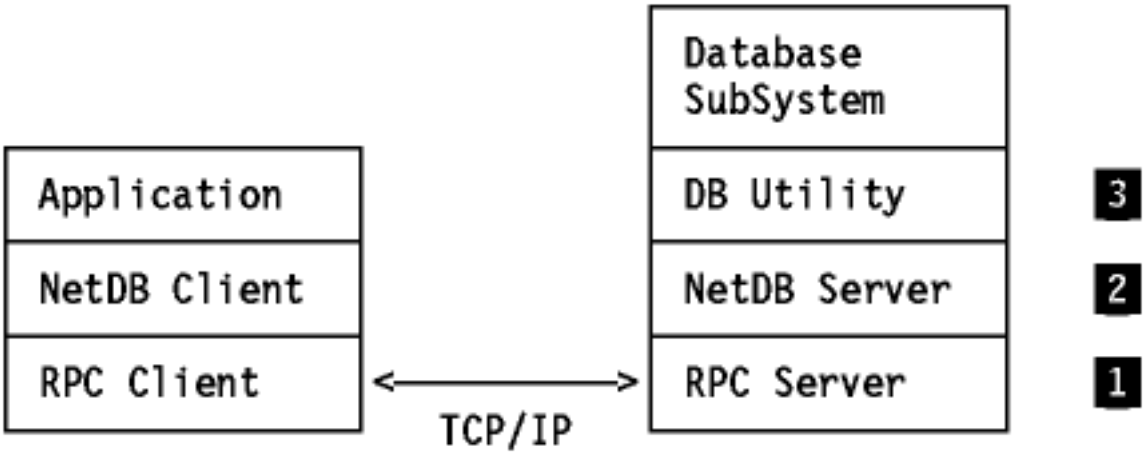


Figura - Componentes del protocolo NDB("Network Database")

1 La capa del **cliente/servidor RPC**
se ocupa del transporte de datos a través de redes TCP/IP: creación de zócalos, interfaces con PORTMAP, enviar y recibir datos.
2 La capa del **cliente/servidor NetDB**
gestiona Unidades de Trabajo("Units of Work"; UOWs), multi - hilos, además de conversión de datos del/al formato ASN.1("Abstract Syntax Notation 1", estándar ISO 8824).
3 La capa de la **utilidad DB**
sirve de interfaz con la base de datos real.

4.23.2 NIS("Network Information Systems")

El NIS("Network Information Systems") no es un estándar de Internet. Fue desarrollado por Sun Microsystems, Inc. Originalmente se le conocía como las Páginas Amarillas.

NIS es un sistema de bases de datos distribuidas que permite compartir la información del sistema en entornos basados en UNIX o AIX. Ejemplos de esta información son los ficheros /etc/passwd, /etc/group /etc/hosts. NIS tiene las siguientes ventajas:

- Proporciona un espacio de nombres consistente para los identificadores de usuario y grupo en un elevado número de

sistemas.

- Reduce el tiempo y esfuerzo del usuario para la gestión de IDs de usuario y grupo, así como de la propiedad de sistemas de ficheros NFS.
- Reduce el tiempo y esfuerzo de los administradores del sistema para la gestión de IDs de usuario y grupo, así como de la propiedad de sistemas de ficheros NFS.

NIS está construido sobre el RPC de Sun. Utiliza el modelo cliente/servidor. Un dominio NIS es una colección de sistemas consistente en:

Servidor NIS maestro
mantiene *mapas*, o bases de datos, que contienen información del sistema tal como passwords y nombres de hosts.
Servidor(es) NIS esclavo(os)
se pueden definir para aliviar al servidor NIS maestro de su carga de trabajo o cuando este no está disponible.
Cliente(s) NIS
son el resto de los sistemas servidos por servidores NIS.

Los clientes NIS no mantienen mapas NIS; interrogan a los servidores NIS acerca de la información del sistema. Cualquier cambio en el mapa NIS de un sistema se efectúa sólo en el servidor NIS maestro(vía RPC), que a su vez propaga los cambios a los servidores NIS esclavos.

Notar que la velocidad de una red determina el rendimiento y disponibilidad de los mapas NIS. Al usar NIS, se debería ajustar el número de esclavos con el fin de optimizar estos parámetros.

4.23.3 Interfaz de zócalos CICS

El CCICS("Customer Information Control System") es un sistema de procesamiento de alto rendimiento. Se desarrolló en IBM.

CICS es el sistema OLTP("Online Transaction Processing") más usado del mercado en la actualidad. Proporciona a los programas de transacciones para comunicaciones de datos(con SNA) y bases de datos (con VSAM, IMS o DB2).

Dada la necesidad de interoperabilidad entre protocolos de red heterogéneos, existe el requerimiento de mejorar la interfaz CICS de comunicación de datos para incluir soporte para TCP/IP además de SNA.

4.23.7 RFC 1006

Los programas escritos originalmente para el XTI("X/Open Transport Interface") pueden usarse con la pila TCP/IP. El RFC 1006 define un componente para el mapeo entre protocolos de modo que estos programas se puedan utilizar en un red TCP/IP.

 Tabla de contenidos  Sinopsis

4.24 Sinopsis

4.24.1 Relaciones cliente/servidor

La siguiente figura muestra los sistemas operativos en los que se implementan aplicaciones TCP/IP.

	S/370			PS/2		RISC/6000	AS/400
	MVS	VM	AIX	DOS	OS/2	AIX	OS/400
FTP	c/s	c/s	c/s	c/s	c/s	c/s	c/s
TELNET	c/s	c/s	c/s	c/	c/s	c/s	c/s
TN3270	c/s	c/s	c/	c/	c/	c/	c/s
SMTP	c/s	c/s	c/s	c/s	c/s	c/s	c/s
SUN RPC	c/s	c/s	c/s	c/	c/s	c/s	c/s
NFS V2	/s	/s	c/s4	c/	c/s	c/s 4	c/s
NCS	c/s	c/s			c/s	c/s	
X Window	c/	c/	c/		c/s	c/s	
REXEC	c/s	c/s	c/s	c/	c/s	c/s	
TFTP		c/	c/s	c/s	c/s	c/s	
LPR/LPD	c/s	c/s	c/s	c/s	c/s	c/s	c/s
SNMP	m/a	m/a		m/a	m*/a	m/a	/a
Sockets	c/s	c/s	c/s	c/s	c/s	c/s	c/s
Kerberos	c/s	c/s			c/s		
DNS	r/s	r/s	r/s	r/	r/s	r/s	r/
TALK			c/s		c/s	c/s	
Finger			c/s	c/	c/	c/s	
PING	x	x	x	x	x	x	x
NETSTAT	x	x	x	x	x	x	x
RIP	x	x	x	x	x	x	

4 = support SUN PC-NFS 4.0
c/s = client/server support

```
4    = support SUN PC-NFS 4.0
c/s  = client/server support
m/a  = monitor/agent support, monitor for DOS: NetView for Windows
r/s  = resolver/server support
*    = with NetView for OS/2
SOD  = Statement of Direction for client/server
cSOD = Statement of Direction for client support
sSOD = Statement of Direction for server support
aSOD = Statement of Direction for agent support
x    = noted function exists for the product
```

Figura: Relaciones cliente/servidor

4.24.2 APIs según el sistema operativo

La siguiente figura muestra los sistemas operativos en los que se implementan APIs para TCP/IP.

	S/370			PS/2		RISC/6000	AS/400
	MVS	VM	AIX	DOS	OS/2	AIX	OS/400
TCP,UDP	C&P	C&P	x			x	x
IP	C&P	C&P	x			x	
FTP				C	C		
SNMP DPI		C			C		
Sockets	M	C	x	C	C	x	x
RPC	C	C	x	C	C	x	x
NCS	C	C			C	x	
X-Window	C	C	W		W	x	
OSF/Motif	C	C	x	C/W	C	x	x
Kerberos	C	C			C	x	
Source avail.	x	x			x		

```
SOD  = Statement of Direction
C    = C Language
C/W  = C Language with Windows
P    = Pascal Language
x    = noted function exists for the product
W    = Athena Widget library
M    = CICS, IMS, REXX, RFC 1006 Sockets
```

Figura: APIs para TCP/IP

 Tabla de contenidos  APIs según el protocolo

4.25 APIs según el protocolo

El siguiente esquema muestra el modelo por capas de la pila de protocolos TCP/IP y además indica las APIs disponibles al usuario.

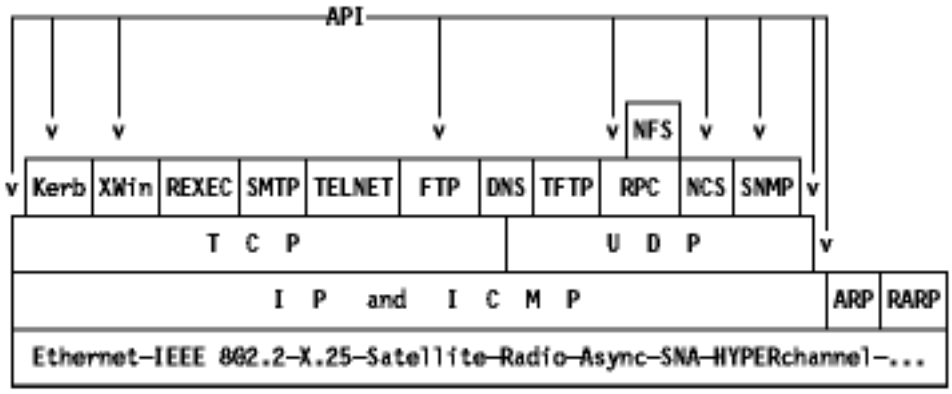


Figura: modelo por capas de TCP/IP

RPC utiliza tanto TCP como UDP. Se ha asociado a UDP debido a que NFS sólo usa RPC sobre UDP.

ARP y RARP sólo se usan en LANs.

La API de zócalos para IP/TCP/UDP ya se ha mencionado en Puertos y zócalos. Las otras APIs ya se han explicado anteriormente en los capítulos:

- Kerberos (ver Sistema de autentificación y autorización Kerberos)
- X Window (ver Sistema X Window)
- FTP (ver FTP("File Transfer Protocol"))
- RPC (ver RPC("Remote Procedure Call"))
- NCS (ver NCS("Network Computing System"))
- SNMP DPI (ver Gestión de red)
- Interfaz de zócalos CICS(ver Interfaz de zócalos CICS)

Tabla de contenidos APIs por protocolo

Capítulo 5. Conectividad

Este capítulo describe diversas opciones de conectividad.

Tabla de contenidos FDDI

5.7 FDDI

Las especificaciones de FDDI definen una familia de estándares para LANs de fibra óptica a 100 Mbps que constituye la capa física y control de acceso al medio según lo define el modelo OSI de ISO.

IP-FDDI es un *borrador* con status *electivo*. Define el encapsulamiento de los datagramas IP y las solicitudes ARP, así como de las respuestas en tramas FDDI. Figura - IP y ARP sobre FDDI muestra las capas de protocolos relacionadas.

Se define en el *RFC 1188 - un estándar propuesto para la transmisión de datagramas IP sobre redes FDDI* para estaciones MAC simples. La descripción de su funcionamiento en estaciones MAC duales está por aparecer en RFC.

RFC 1188 afirma que todas las tramas se transmiten en el formato estándar "IEEE 802.2 LLC Type 1 Unnumbered Information", con los campos DSAP y SSAP de la cabecera 802.2 fijados al valor global SAP asignado a SNAP(170, en decimal). El código de 24 bits "Organization Code" en la cabecera SNAP se pone a cero, y los restantes 16 bits son el "EtherType", es decir:

- 2048 para IP
- 2054 para ARP

El mapeado de las direcciones de 32 bits de Internet a las direcciones de 48 bits FDDI se realiza mediante el procedimiento de descubrimiento dinámico de ARP. Las direcciones de broadcast de Internet(con la dirección de host toda a unos) se mapean a direcciones de broadcast FDDI(toda a unos).

Los datagramas IP se transmiten como series de bytes de 8 bits empleando el orden de transmisión TCP/IP "big-endian" o "network byte ".

La especificación de FDDI MAC (*ISO 9314-2 - ISO, Interfaz de datos distribuida de datos para fibra - control de acceso al medio*) define un tamaño máximo de 4500 bytes para todos los campos de la trama. Después de tener en cuenta la cabecera LLC/SNAP, y de permitir futuras extensiones a la cabecera MAC y a los campos de estado de la trama, la MTU de las redes FDDI es de 4532 bytes.

Remitirse a *Conceptos y productos de LANs,* GG24-3178 para más detalles acerca de la arquitectura FDDI.

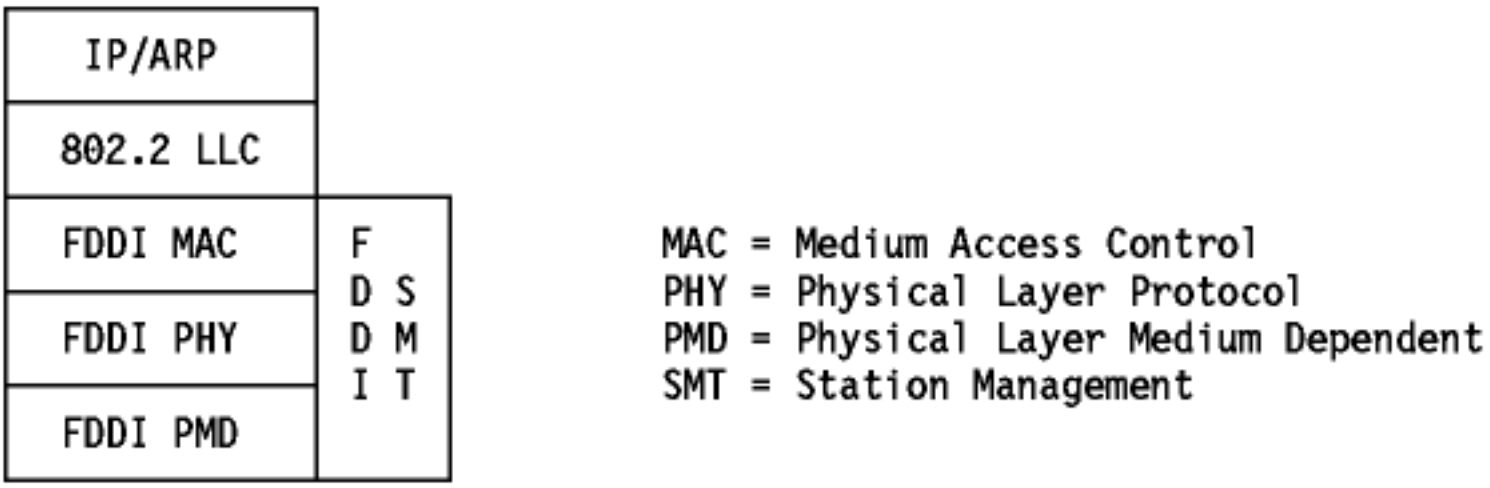


Figura: IP y ARP sobre FDDI

5.8 SLIP("Serial Line IP")

la familia de protocolos TCP/IP se ejecuta sobre una variedad de redes: IEEE 802.3 y LANs 802.5, líneas X.25, enlaces por satélite, y líneas en serie. Los estándar para la encapsulación de paquetes IP se han definido para muchas de estas redes, pero no hay un estándar para las líneas serie. SLIP es un estándar *de facto* actual, usado habitualmente para conexiones punto-a-punto serie con TCP/IP. No es un estándar de Internet.

SLIP es un protocolo muy simple diseñado hace mucho tiempo y es meramente un protocolo de entramado de paquetes. Define una secuencia de caracteres que sirven de trama a los paquetes IP en una línea serie. No proporciona:

- Direccionamiento: ambos ordenadores en un enlace SLIP necesitan conocer la dirección del otro para el encaminamiento.
- Identificación del tipo de paquete: así, sólo se puede ejecutar un protocolo en una conexión SLIP.
- Detección/corrección de error: la detección de error no es absolutamente necesaria en el nivel de SLIP ya que cualquier aplicación IP debería detectar paquetes corruptos(la cabecera IP y los checksums UDP/TCP deberían bastar). Como lleva bastante tiempo retransmitir un paquete alterado, se ganaría en eficiencia si SLIP pudiera proporcionar algún tipo sencillo de mecanismo de corrección de error propio.
- Compresión.

Se espera que el protocolo PPP("Point-to-Point Protocol"). Remitirse a PPP("Point-to-Point Protocol").

5.8.1 Ejemplo

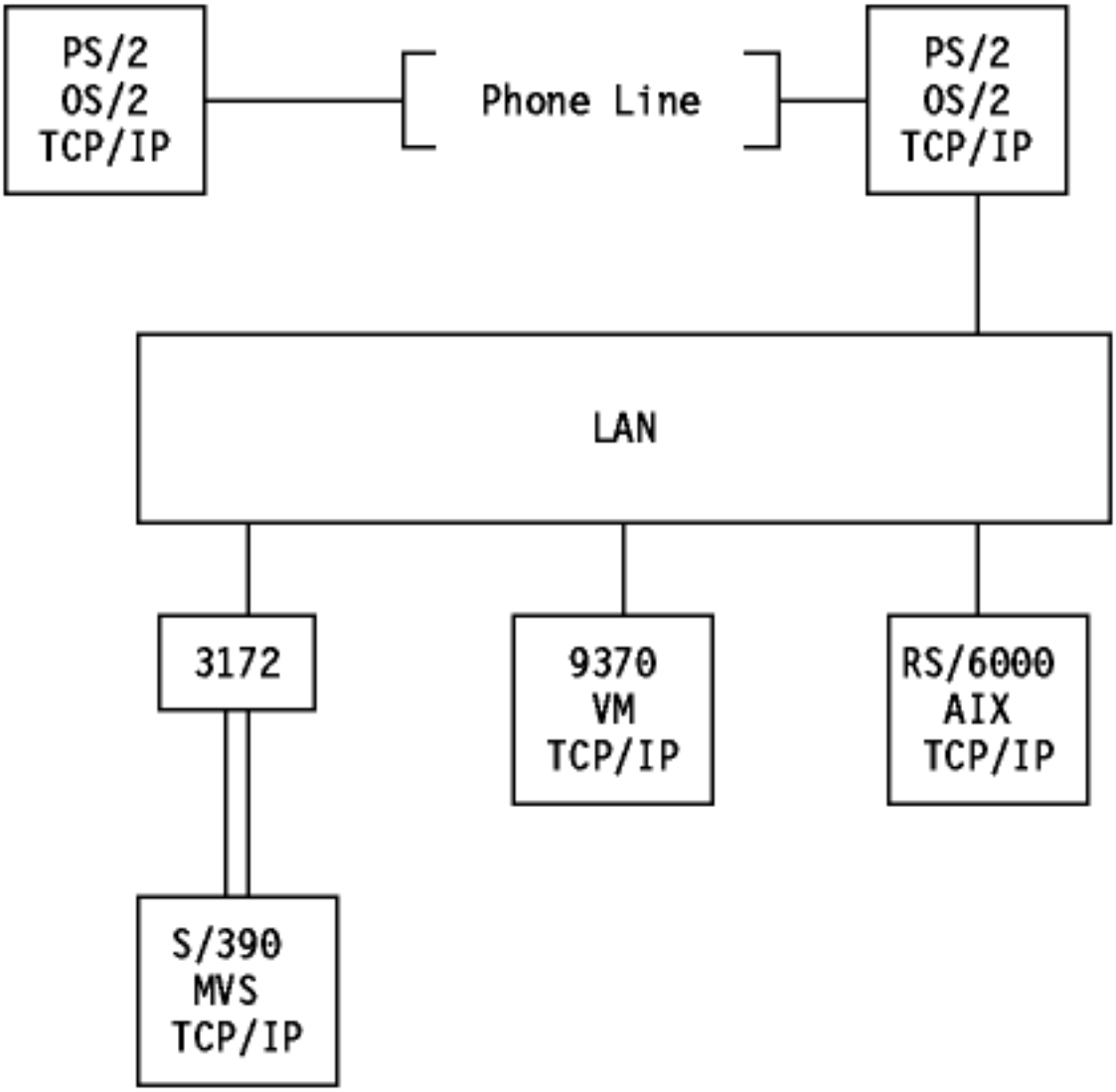


Figura: ejemplo de SLIP

En Figura - ejemplo de SLIP, la estación de trabajo OS/2, conectada a la LAN con una conexión SLIP, puede acceder al resto de las estaciones de trabajo, asumiendo que la información de encaminamiento necesaria ya ha sido determinada. Inversamente, todos los hosts conectados a la LAN puede acceder a ella.

Tabla de contenidos PPP("Point-to-Point Protocol")

5.9 PPP("Point-to-Point Protocol")

PPP es un *protocolo estándar específico de red* con STD 51. Su status es *electivo*. Se describe en el RFC 1661 y en el RFC 1662.

Hay un gran número de *propuestas de estándares* que especifican el funcionamiento de PPP sobre distintas clases de enlaces punto-a-punto. Cada uno tiene un status electivo. El lector puede consultar el *STD 1 - Estándares de protocolos oficiales de Internet* para una lista de RFCs relacionados con PPP contemplados en el seguimiento de estándares("Standards Track").

Los circuitos punto-a-punto en la forma de líneas síncronas y asíncronas han sido durante mucho tiempo la vela mayor de las comunicaciones de datos. En el mundo TCP/IP, SLIP ha tenido una gran utilidad en esta área, y aún está ampliamente extendido en conexiones TCP/IP con marcaje. Sin embargo, SLIP tiene una serie de desventajas:

- SLIP define sólo el protocolo de encapsulamiento, pero no ninguna forma de negociación o control del enlace. Los enlaces se establecen y configuran manualmente, incluyendo la especificación de la dirección IP.
- SLIP se define sólo para enlaces asíncronos.
- SLIP no puede soportar múltiples protocolos sobre un solo enlace; todos los paquetes deben ser datagramas IP.
- SLIP no realiza ningún tipo de detección de error lo que obliga a que los protocolos de niveles superiores tengan que retransmitir en caso de error o de líneas con ruido.
- SLIP no aporta mecanismos de compresión para los campos de cabecera IP usados con frecuencia. Muchas aplicaciones sobre enlaces serie lentos tienden a generar tráfico TCP interactivo para un solo usuario, como por ejemplo TELNET. Esto suele implicar tamaños de paquete pequeños y por lo tanto una cantidad de overhead relativamente grande en las cabeceras TCP e IP que no cambian mucho entre datagramas, pero que producen un detrimento en los tiempos de respuesta de tipo interactivo.

  Sin embargo, muchas implementaciones de SLIP usan ahora la *compresión de cabecera Van Jacobsen*, con el fin de reducir el tamaño de las cabeceras TCP e IP de 40 bytes a 8 bytes, guardando los estados de un conjunto de conexiones TCP en cada extremo del enlace y reemplazando las cabeceras con actualizaciones codificadas, en el caso habitual en el que muchos de los campos cambian poco o nada entre sucesivos datagramas IP en una sesión. El RFC 1144 describe esta compresión.

PPP maneja estos problemas.

PPP tiene tras componentes principales:

1. Un método para encapsular datagramas sobre enlaces serie.
2. Un LCP(*"Link Control Protocol")* para establecer, configurar y probar la conexión
3. Una familia de NCPs(*"Network Control Protocol")* para establecer y configurar distintos protocolos de red. PPP está diseñado para permitir el uso simultáneo de múltiples protocolos de red.

Antes de que un enlace se considere apto para que los protocolos de red lo utilicen, debe producirse una secuencia específica de eventos. El LCP proporciona un método para establecer, configurar, mantener y terminar la conexión. LCP atraviesa las siguientes fases:

1. Establecimiento del enlace y negociación de la configuración:

   En esta fase, se intercambian paquetes de control del enlace y se negocian opciones de configuración. Una vez que hay acuerdo sobre las opciones, el enlace se *abre*, pero no está necesariamente listo para que los protocolos de red comiencen a funcionar.

2. Determinación de la calidad del enlace:

   Esta fase es opcional. PPP no especifica una política para determinar la calidad, pero proporciona herramientas de bajo nivel, tales como "echo request" y "echo reply".

3. Autentificación:

   Esta fase es opcional. Cada extremo del enlace se autentifica con el otro extremo empleando métodos de autentificación acordados durante la fase 1.

4. Negociación de la configuración del protocolo de red:

   Una vez que LCP ha terminado la fase anterior, los protocolos de la capa de red se pueden configurar por separado con el NCP apropiado.

5. Terminación del enlace:

   LCP puede terminar el enlace en cualquier momento. Esto se hará normalmente a petición de un usuario humano, aunque puede ocurrir debido a un evento físico.

El *IPCP("IP Control Protocol")* es el NCP para IP y es responsable de configurar, habilitar y deshabilitar el protocolo IP en ambos extremos del enlace. La secuencia de negociación de opciones es la misma que en LCP, lo que da la posibilidad de reutilizar el código.

Una opción importante usada con IPCP es *la compresión de cabecera Van Jacobsen*, empleada para reducir el tamaño combinado de las cabeceras IP y TCP de 40 bytes a aproximadamente 4 byte, guardando los estados de un conjunto de conexiones TCP en cada extremo del enlace y reemplazando las cabeceras con actualizaciones codificadas, en el caso habitual en el que muchos de los campos cambian poco o nada entre sucesivos datagramas IP en una sesión. El RFC 1144 describe esta compresión.

 Tabla de contenidos  Acceso a Internet

Capítulo 6. Acceso a Internet

En el pasado, Internet estaba reservado para que los investigadores, científicos y académicos de todo el mundo intercambiaran o publicaran información. Las principales herramientas de comunicación en la red eran el E - Mail, FTP y TELNET. El acceso a Internet para particulares era muy difícil y su uso con fines comerciales estaba estrictamente prohibido debido a que Internet estaba subvencionada en gran parte por el gobierno. La capacidad del hardware(terminales no programables), las velocidades de los módems, y el ancho de banda de la red hacían la presentación gráfica imposible.

Gradualmente, Internet se fue abriendo a todo el mundo, gracias a las compañías proveedoras que suministraban acceso a Internet a un nivel asequible para particulares. Este proceso se debió a una caída en los costes de la tecnología, incluyendo los servicios de telecomunicación, haciendo por tanto más económico el acceso a Internet. La necesidad de encontrar verdaderas herramientas de navegación para explorar Internet está asociada a este cambio. Las metas de un sistema con acceso a Internet son:

- Facilidad de navegación y uso
- Soporte a la navegación
- Proporcionar acceso a tecnologías complejas de Internet, tales como TELNET, Gopher, USENET, FTP, mail, etc.
- Independencia del sistema operativo
- Acceso simple texto, audio, vídeo y gráficos(presionando sólo un botón)

Toda una variedad de navegadores de Internet están disponibles. Cada herramienta es particularmente adecuada para ciertos tipos de aplicación. Estas son las más populares, brevemente descritos:

- WAIS("Wide Area Information Services")

  Muy apropiado para indexar y buscar en grandes bases de datos. Ampliamente usada por bibliotecarios e investigadores.

- Gopher

  Herramienta que proporciona un acceso por menú a información existente. Muchas organizaciones lo usan en vez del FTP anónimo. Ver Gopher.

- Veronica:

  Herramienta de búsqueda para el entorno Gopher. Ver Veronica.

- Archie

  Uno de los primeros navegadores, usado para buscar ficheros en sitios FTP.

- WWW("World Wide Web")

  Proporciona un índice de la información existente como el Gopher, pero soporta hipertexto e hipermedia para la creación de nuevos documentos que contengan texto, audio, vídeo y gráficos. WWW se ha hecho muy popular. Ver WWW("World Wide Web").

El concepto de *cortafuegos("firewall")* se describe posteriomente en este capitulo.

6.1 Gopher

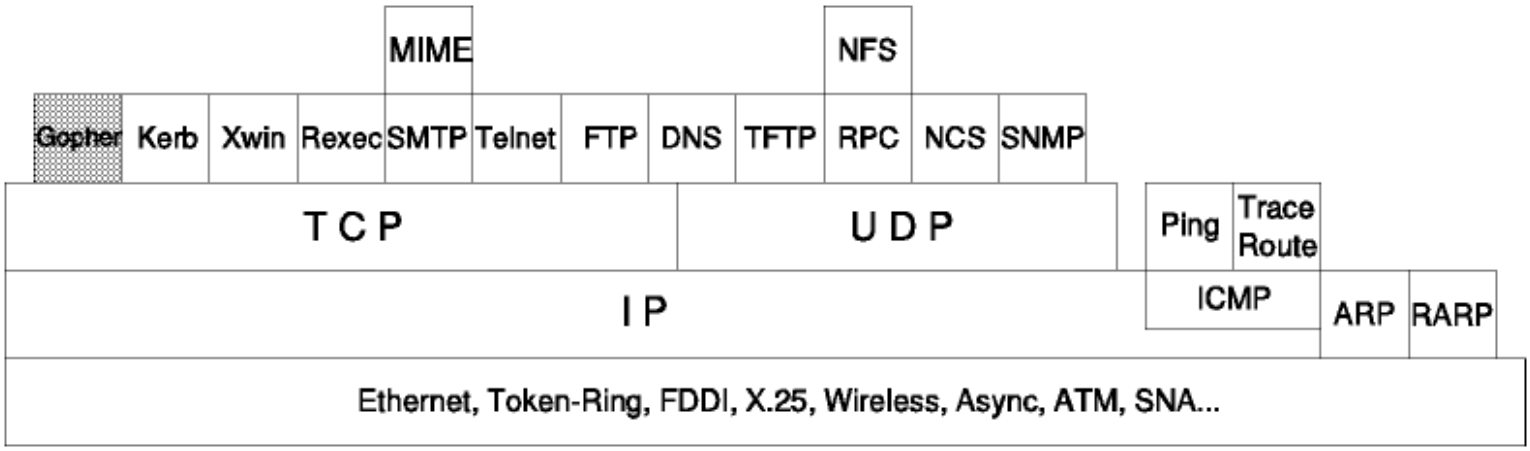


Figura: Gopher

En pocas palabras, el Gopher es un sistema distribuido de búsqueda y recuperación de documentos. Combina las mejores características de exploración a través de colecciones de información y de bases de datos completamente indexadas. El protocolo y el software siguen un modelo cliente/servidor y permiten a los usuarios de los más variados sistemas de sobremesa explorar, buscar y recuperar documentos residentes en múltiples servidores distribuidos.

El protocolo Gopher se desarrolló en la Universidad de Minesota y está disponible en el RFC 1436. Su estado es *informacional*.

La razón de desarrollar el Gopher fue la necesidad de un sistema de información a nivel de campus que permitiese a cualquiera publicar documentos o información incluso con un pequeño ordenador de sobremesa. El software de clientes Gopher concede a los usuarios una jerarquía de ítems y directorios muy parecida a un sistema de ficheros. De hecho, la interfaz del Gopher está diseñada para asemejarse a un sistema de ficheros, ya que un sistema de ficheros es un buen modelo para localizar documentos y servicios. De este modo, al conectarse a un servidor Gopher se tiene una lista de distintos ítems, similar a la apariencia de directorio raíz de un PC. Después de seleccionar un ítem del menú, se obtienen todos los ítems incluidos en él, de forma similar a un subdirectorio, y así sucesivamente. Si se selecciona un ítem que representa un fichero, el fichero se transmite automáticamente y se le muestra al cliente. No es necesario que el fichero esté localizado en el mismo servidor Gopher donde se obtuvo la información del sistema de archivos. Un menú Gopher puede incluir ítems de servidores Gopher distintos y el usuario se conecta de modo automático al servidor donde se halla el ítem seleccionado. El usuario no sabe ni le importa el hecho de que los ítems a seleccionar puedan residir en muchas máquinas distintas en Internet.

Un ejemplo sencillo:

Te gustaría echarle un vistazo al menú de la cafetería

1. Arranca el cliente Gopher y conéctate al servidor Gopher principal
2. Localiza un ítem en el que pueda estar el menú y selecciónalo. Por ejemplo: Servicios
3. Se muestran los ítems de Servicios
4. Uno de estos ítems puede ser "Menú del día", que es un fichero que se halla en el servidor Gopher de la cafetería
5. Si lo has seleccionado, el cliente Gopher establece automáticamente una conexión con el servidor al que apunta el ítem, lo recupera y muestra el fichero

Puede que la ruta al archivo sea muy compleja, y que muchos servidores Gopher de todo el mundo estén implicados

Para implementar la jerarquía mencionada anteriormente, el cliente Gopher necesita alguna información acerca del tipo de objeto con el fin de mostrar el ítem que representa el fichero o el directorio, por ejemplo. El tipo de Gopher se codifica con un sólo dígito al comienzo de cada línea. La siguiente es una lista de tipos conocidos del Gopher que se incluyen en el RFC:

- 0 - Fichero
- 1 - Directorio
- 2 - CSO (qi) servidor de guía telefónica
- 3 - Error
- 4 - Archivo BinHex de Macintosh
- 5 - Archivo binario de algún tipo de DOS
- 6 - Fichero UNIX con codificación uu
- 7 - Servidor de búsqueda e indexación
- 8 - Sesión de telnet orientada a carácter
- 9 - Fichero binario
- T - Conexión TN3270
- s - Tipo de sonido. El flujo de datos es sonido mulaw
- g - Tipo GIF
- M - Datos en formato MIME
- h - Tipo html
- I - Tipo imagen
- i - Tipo texto "inline"

Los siguientes párrafos describen el funcionamiento básico del protocolo Gopher

En esencia, el protocolo Gopher consiste en una conexión de un cliente a un servidor y en enviar al servidor un selector(una línea de texto, que puede estar vacía) por medio de una conexión TCP/IP. El servidor responde con un bloque de texto terminado en una línea que contiene un punto, y cierra la conexión. El servidor no retiene información de estado entre transacciones con un cliente. Asumir, por ejemplo, que un servidor Gopher escucha en el puerto 70. La única información de configuración que el software del cliente retiene es el nombre del servidor y el número de puerto(en este ejemplo la máquina es rawBits.micro.umn.edu y el puerto el 70). En el ejemplo de abajo el carácter F denota al carácter TAB.

```
Cliente: {Abre la conexión con rawBits.micro.umn.edu en el puerto 70}
Server: {Acepta la conexión pero no dice nada}

Client: <CR><LF> {Envía una línea vacía, queriendo decir: "Muestra lo que tienes"}
Server: {Envía una serie de líneas, cada una acabada en CR LF}
0About internet GopherFStuff:About usFrawBits.micro.umn.eduF70
1Around University of MinnesotaFZ,5692,AUMFunderdog.micro.umn.eduF70
1Microcomputer News & PricesFPrices/Fpserver.bookstore.umn.eduF70
1Courses, Schedules, CalendarsFFevents.ais.umn.eduF9120
1Student-Staff DirectoriesFFuinfo.ais.umn.eduF70
1Departmental PublicationsFStuff:DP:FrawBits.micro.umn.eduF70
                  {.....etc.....}
.                 {$Period on a line by itself$}
                  {El servidor cierra la conexión}
```

El primer carácter de cada línea describe el tipo de Gopher como se muestra más arriba. Los siguientes caracteres hasta llegar a un TAB constituyen la ristra de caracteres a mostrar al usuario para que haga su selección. Los caracteres que sigan a ese TAB hasta llagar al siguiente forma una ristra selectora que el cliente debe enviar al servidor para recuperar el documento(o el listado de un directorio). En la práctica, la ristra selectora suele ser una ruta de acceso u otro selector de un fichero que permita al servidor localizar el ítem deseado. Los dos siguientes campos delimitados por TAB denotan el nombre del dominio del host o directorio que posee el documento, y el puerto con el que conectarse. Un CR/LF denota el fin del ítem. El cliente puede presentar el flujo de datos descrito arriba del modo siguiente:

```
About Internet Gopher
Around the University of Minnesota...
Microcomputer News & Prices...
Courses, Schedules, Calendars...
Student-Staff Directories...
Departmental Publications...
```

En este caso, los directorios se muestran con una elipsis, y los ficheros sin ninguna. Sin embargo, dependiendo de la plataforma para la que se ha diseñado el cliente y del gusto del autor, los ítems se pueden representar con otras marcas de texto o con iconos.

En el ejemplo, la línea 1 describe un documento que el usuario verá como *"About Internet Gopher"*. Para recuperar este documento, el software cliente debe enviar la cadena: *"Stuff:About us" to rawBits.micro.umn.edu at port 70*. Si el cliente hace esto, el servidor responderá con los contenidos del documento, terminados en una línea. Como se puede ver en el ejemplo, el usuario no conoce o no le importa que los ítems a seleccionar pueden hallarse en distintas máquinas de Internet. La conexión entre servidor y cliente sólo existe mientras la información es transferida. Después de esto el cliente se puede conectar con un servidor diferente con el fin de conseguir los contenidos de un directorio dado.

Para más información acerca del protocolo Gopher, remitirse al *RFC 1436*. Para una lista de FAQs , incluyendo los FTPs anonónimos para obtener el código del cliente y el servidor Gopher, descargar el siguiente fichero del FTP anónimo: URL:ftp://rtfm.mit.edu/pub/usenet/news.answers/gopher-faq.

6.1.2 Veronica

A pesar de lo interesante que puede ser explorar el "Gopherespacio", puede que un día quieras recuperar alguna información o un fichero de un servidor Gopher. El problema es como acceder a los servidores adecuados con la información necesaria sin tener que llamar a interminables menús Gopher. Afortunadamente, hay una forma de hacer el Gopher aún más sencillo de usar.

Esta herramienta se llama Veronica("Very Easy Rodent-Oriented Net-wide Index to Computerized Archives") y hace por el Gopherespacio lo que *Archie* por los sitios FTP.

Veronica es un sistema de localización de recursos que proporciona acceso a información de recursos que tienen la mayoría (99% o más) de los servidores Gopher de todo el mundo. Además de los datos del Gopher nativo, Veronica incluye referencias a muchos recursos suministrados por otros tipos de servidores de información, como servidores WWW, archivos usenet, y servicios de información accesible por telnet.

Las consultas de Veronica son búsquedas que manejan una clave a buscar en un título de un ítem. Una simple consulta puede ser muy potente porque un elevado número de servidores de información se incluyen en el índice.

El acceso a Veronica se efectúa por medio de clientes Gopher. Un usuario de Veronica envía una consulta(a través de un cliente Gopher), que puede contener una expresión clave booleana además de directivas especiales de Veronica. El resultado de la búsqueda de Veronica es un menú Gopher que incluye los ítems de información cuyos títulos contienen la clave especificada. Estos menús se pueden explorar como cualquier otro menú de Gopher.

En enero de 1995, estaban indexados 5057 servidores Gopher. El índice incluye también ítems de aproximadamente otros 5000 servidores, en los casos donde estos últimos son referenciados desde menús del Gopher. Estos otros servidores consisten principalmente en 3905 servidores WWW y alrededor de 1000 servicios de tipo TELNET.

Veronica se puede encontrar en la mayoría de los servidores Gopher, seleccionando "Other Gopher and Information Services" en el menú principal y luego "Searching through Gopherspace using Veronica". Si el servidor Gopher no proporciona estos servicios, es posible conectarse directamente a Veronica via Gopher://veronica.scs.unr.edu.:70/11/veronica. Allí se puede encontrar además información adicional sobre Veronica.

 Tabla de contenidos  WWW ("World Wide Web")

6.2 WWW("World Wide Web")

El World Wide Web es un sistema gobal de hipertexto desarrollado inicialmente en 1989 por Tim Berners Lee en el Laboratorio Europeo de Física de Partículas, ("European Laboratory for Particle Physics, CERN") en Suiza. En 1993 el Web comenzó a crecer rápidamente, principalmente gracias a la NCSA ("National Center for Supercomputing Applications"), que desarrolló un navegador Web llamado Mosaic, una aplicación basada en X Windows. Esta aplicación proporcionó la primera interfaz gráfica de usuario al Web, haciendo la navegación más asequible

Hoy en día existen navegadores y servidores Web disponibles para casi todas las plataformas. Se pueden conseguir gratis, vía FTP, o comprando una copia con licencia. El rápido crecimiento de la popularidad del Web se debe a la flexibilidad con la que la gente puede navegar a través de recursos de todo el mundo en Internet, así como descargarlos de la red. Para hacerse una idea del crecimiento del Web, aquí hay algunas estadísticas:

- Junio de 1993 - sólo 130 sitios Web disponibles
- Diciembre de 1994 - más de 11500 sitios Web disponibles

El número de servidores Web está creciendo con gran rapidez(entre 50 y 100 cada día) y el tráfico en el puerto 80, que es el puerto Web *bien conocido*, de la troncal NSF experimenta igualmente un crecimiento fenomenal.

Ya hay bastantes compañías haciendo negocio en el Web. Se pueden encontrar prospectos ofertas de productos, y por supuesto, pedidos, en todo el Web. La mayoría de las multinacionales tienen un servidor Web para distribuir información específica de sus productos, documentos o simplemente para ponerse en contacto con clientes. Una *página* es sólo el término Web para un documento y la *página particular/personal* un punto de partida para una colección de documentos. Es, si se desea, la tabla de contenidos de un sitio Web. Desde allí se puede explorar y buscar con facilidad en todo el Web.

Presentar un documento en forma de hipertexto tiene ciertas ventajas para el usuario. Por ejemplo, si se quiere más información acerca de un tema concreto que se haya mencionado, suele ser posible hacer click sobre él para leer más detalles al respecto. Los temas con un enlaces a otro documento se pueden identificar fácilmente al estar resaltados. De hecho, los documentos pueden ser y son enlazados con frecuencia a otros documentos de autores completamente distintos, como si fueran notas a pie de página, pero en las que el gráfico o documento se muestra inmediatamente. Un documento en el Web podría incluir enlaces a otros documentos localizados en diferentes sitios Web. Al activar el enlace, casi siempre con un click del ratón, el documento se recupera del correspondiente servidor y se muestra automáticamente. Este documento podría a su vez incluir enlaces a otros recursos, y así sucesivamente.

El protocolo estándar de comunicaciones entre servidores y clientes Web es el HTTP("Hypertext Transfer Protocol"), que es un borrador de estándar de Internet. El HTTP es un protocolo orientado a objetos genérico y sin estado. El IETF ha establecido un grupo de trabajo para mejorar su eficacia. Los navegadores pueden usar además otros protocolos como el FTP, Gopher, WAIS y NNTP ("Network News Transfer Protocol") por ejemplo. Por ello, no hace falta un cliente determinado para conseguir acceso a todos estos otros recursos que también están disponibles en la red. El modo en que los navegadores pueden diferenciar entre todos estos protocolos y qué protocolos son los que soportan se explica posteriormente en esta sección.

Una transacción HTTP consiste básicamente en:

Conexión
: El establecimiento de una conexión del cliente con el servidor. El puerto TCP/IP 80 es el puerto bien conocido, pero el URL puede especificar otros puertos no reservados.

Solicitud
: El envío por parte del cliente de un mensaje de solicitud al servidor.

Respuesta
: El envío por parte del servidor de una respuesta al cliente.

Cierre
: El cierre de la conexión por parte del cliente y el servidor.

Para una descripción más detallada de HTT, remitirse a los documentos del grupo de trabajo del IETF.

El lenguaje estándar de marcas para documentos Web es HTML ("Hypertext Markup Language"), que es un borrador de estándar de Internet y actualmente varios grupos de trabajo del IETF están trabajando en él. HTMP es una aplicación de

SGML("Standard Generalized Markup Language"). Para crear un documento Web hay que usar las marcas HTML que constituyen la estructura lógica del documento, por ejemplo, cabeceras, listas y párrafos. Aquí se muestran algunas marcas para definir enlaces a otros documentos o para embeber una imagen en el texto.

```
<HTML>  <!-- Begin of document -->
 <HEAD>  <!-- A sample document -->
  <TITLE>This is a Sample</TITLE>
 </HEAD>  <!-- End of the heading section -->
 <BODY>  <!-- Begin of text body -->
  <H1>First Header</H1>
   <P>The first paragraph.
   <UL>  <!-- unordered list -->
    <LI>Item one
   </UL> <!-- End of list -->
 </BODY> <!-- End of text body -->
</HTML> <!-- End of document -->
```

Si quieres iniciarte en HTML, puedes ver el siguiente documento: http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html.

Todos los documentos, imágenes, clips de audio o de vídeo se denomina recurso Web Para identificar el método de acceso a estos recursos el Web emplea URLs("(Uniform Resource Locators). URL es un protocolo estándar de Internet y se puede encontrar en el RFC 1738. El contexto global para construir nuevos esquemas para codificar nombres y direcciones de objetos en Internet se describe en el RFC informacional 1630. Este RFC acuña el término URI(Universal Resource Identifiers) como un modelo más teórico para diseñar estos esquemas. Los URIs que se refieren a una dirección objeto(dirección IP e información de la ruta de acceso)mapeados a un método de acceso conocido usando un protocolo de red existente como HTTP o FTP se conocen como URLs. Por lo tanto, un URL es una forma específica de un URI. En general, los URLs se escriben del modo siguiente:

```
<scheme>:<scheme-specific-part>
```

Un URL contiene el nombre del esquema usado(<scheme>) seguido de una coma y una ristra(<scheme-specific-part>) cuya interpretación depende del esquema. Los sigs esquemas los cubre el RFC, y les pueden seguir otros en el futuro:

- ftp - "File Transfer protocol"
- http - "HyperText Transfer Protocol"
- gopher - El protocolo Gopher
- mailto - Dirección de Correo Electrónico
- news - "USENET news"
- nntp - "USENET news" usando acceso NNTP
- telnet - Sesiones interactivas
- wais - "Wide Area Information Servers"
- file - Nombres de fichero específicos de un host
- prospero - "Prospero Directory Service"

Mientras que la sintaxis para el resto del URL puede variar dependiendo del esquema seleccionado, los esquemas que implican el uso directo de un protocolo basado en IP usan una sintaxis común para la parte <scheme-specific data>, que comienza por "//" para indicar que sigue la sintaxis estándar de Internet:

```
//<user>:<password>@<host>:<port>/<url-path>
```

Algunas o todas de las partes "<user>:<password>@", ":<password>", ":<port>", and "/<url-path>" se pueden excluir.

El "url-path" al final proporciona detalles de cómo acceder el recurso especificado. Nótese que el "/" entre el host(o puerto) y el "url-path" no forma parte del "url-path".

Según la definición anterior, el URL de HTTP tiene este aspecto:

```
http://<host>:<port>/<path>?<searchpart>
```

Donde:

host
: El nombre de dominio completo de un host o una dirección IP en formato decimal.

port
: El número de puerto al que conectar. Si este parámetro se omite en un URL de HTTP, es 80 por defecto.

path
: Especifica un selector HTTP, una ruta a un documento HTML, por ejemplo.

? searchpart
: Ristra de consulta("query string") indicada con "?".

La sintaxis para los demás esquemas como FTP y Gopher, por ejemplo, se explican en el RFC 1738.

Hay tres formas de acceder a la Web:

- Usar un navegador

  Es la mejor opción, aunque la LAN debe tener acceso a Internet. En la mayoría de los casos estas redes no tienen acceso directo a Internet, sino que se conectan a través de un cortafuegos. En este caso hay que especificar un servidor SOCKS o un proxy en el que el host se registra para obtener el acceso. Otra forma de conectarse es con el protocolo SLIP.

- Usar un navegador en una máquina a la que se tiene acceso por TELNET.
- Acceder la Web por E-mail.

Los navegadores están disponibles para la mayoría de las plataformas. Para conseguir una lista de sitios FTP que los proporcionan y otras informaciones útiles, ir a ftp://rtfm.mit.edu/pub/usenet/news.answers/www/faq.

 Tabla de contenidos  Cortafuegos("Firewalls")

Tabla de contenidos WWW("World Wide Web")

6.3 Cortafuegos("firewalls")

En esencia, un cortafuegos es una barrera entre una red segura, privada e interna y otra red(insegura) en Internet. El propósito de un cortafuegos es prevenir comunicación no deseada o no autorizada con la red segura. Sus tareas son dos:

- Evitar que los usuarios de la propia red intercambien información libremente con usuarios externos a ella
- Evitar la entrada de usuarios externos a la propia red que pretendan atacarla o comprometer su integridad

Normalmente, los host de una red segura no pueden acceder a una red externa. Esto reduce el riesgo de intrusiones por parte de usuarios no autorizados aunque impide la accesibilidad a Internet a los usuarios de esa red. Sin acceso a Internet, los usuarios de una red segura no pueden manejar importantes herramientas, tales como TELNET, FTP, Gopher, y WWW, con el fin de acceder a los recursos disponibles en Internet.

Un cortafuegos puede proteger una red de diversas formas. Puede proporcionar servicios de encubrimiento que nieguen o garanticen los accesos basados en cosas como el nombre del usuario, el nombre del host, y el protocolo TCP/IP. Un cortafuegos puede suministrar también una variedad de servicios que dejen paso a los usuarios autorizados mientras excluyen a los no autorizados. Al mismo tiempo, asegura que todas las comunicaciones entre la red e Internet dan la impresión de finalizar en el cortafuegos, evitando que el mundo externo puede vislumbrar en modo alguno la estructura de la red.

Tabla de contenidos DCE("Distributed Computing Environment")

Apéndice B. DCE("Distributed Computing Environment")

DCE se está convirtiendo en una tecnología muy importante en el desarrollo de aplicaciones distribuidas en entornos abiertos y heterogéneos. Su papel es análogo al que juega TCP/IP en entorno de red distribuidos. En este capítulo, resumimos la historia de DCE y sus principales componentes.

B.1 Historia

OSF("Open Software Foundation") es una organización sin ánimo de lucro para la investigación y el desarrollo que proporciona el software esencial en la creación de entornos de computación en sistemas abiertos. Fundada originalmente por IBM, DEC, Apollo, HP, Groupe Bull, Nixdorf y Siemens, OSF tiene más de 350 miembros en todo el mundo.

OSF ha definido el DCE("Distributed Computing Environment") para simplificar el desarrollo de aplicaciones en entornos heterogéneos.

Fundada en 1984, X/Open es una organización mundial e independiente de sistemas abiertos, dedicada al desarrollo de un CAE("Common Applications Environment") abierto y sin exclusivas de distribución(no tiene por qué haber un distribuidor exclusivo), basado en estándares internacionales y de facto.

En agosto de 1992, X/Open y OSF anunciaron que realizarían un esfuerzo conjunto para integrar el DCE de OSF en el CAE de X/Open.

Tabla de contenidos Historia

B.2 Descripción de los componentes de la tecnología DCE

DCE consta de los siguientes componentes:

- **Hilos DCE**: soporta multihilos en un mismo proceso.
- **RPC DCE**: consiste en una herramienta de desarrollo y una librería runtime. Cuando se utilizan sus características de compatibilidad, es compatible con la versión 1.5.1 de NCS. Sin embargo, futuras mejoras de DCE RPC harán que sean incompatibles.
- **Servicio de directorios DCE**: proporciona un depósito central de información sobre los recursos del entorno distribuido.
- **DTS("Distributed Time Service")**: asegura la sincronización entre los relojes del sistema distribuido.
- **Servicio de seguridad DCE**: proporciona funciones de seguridad basadas en la versión 5 de Kerberos.
- **DFS("Distributed File Service")**: proporciona un sistema avanzado de ficheros basado en el AFS("Andrew File System", un sistema distribuido de ficheros desarrollado originalmente por la Universidad Carnegie-Mellon e IBM, ahora comercializado por Transarc Corporation).
- **Servicio DCE de apoyo a sistemas sin disco**: permite que un sistema sin disco opere en un entorno DCE.

Actualmente DCE se halla en su versión 1.1, lanzada en noviembre de 1994. Algunas de sus nuevas características son:

- Funciones de administrador mejoradas: incluyen una interfaz de usuario común entre todos los componentes DCE, permitiendo arranque remoto, la administración y el cierre de los servicios DCE. También se incluyen mensajes de diagnóstico mejorados que ayudan a solucionar problemas en las redes heterogéneas sobre las que se implementa DCE.
- Mejoras en la seguridad: los sistemas que no están basados en RPC ya pueden usar la seguridad DCE. Además, los procedimientos de auditoría y las funciones de comprobación de passwords y de pre-autentificación han sido mejoradas.
- Soporte del lenguaje nacional: los mensajes DCE se pueden presentar en el lenguaje local y las aplicaciones RPC se pueden configurar para convertir los datos a este lenguaje.
- Incremento del rendimiento: el IDL("Interface Definition Language") produce código más eficiente y con un flujo continuo. Además, RPC ha sido optimizado.
- Pasarela NFS/DFS: permite el acceso de NFS a DFS.

Está previsto se comience a probar la versión 1.2 en el verano de 1995. También se espera que incluya nuevas características administrativas y de interoperabilidad además de funciones de escalabilidad y gestión de redes.

Tabla de contenidos Glosario

Glosario

A

Sintaxis abstracta(abstract syntax)
: Una descripción de estructuras de datos independiente de la máquina.

ACSE: Association Control Service Element
: El método usado en OSI para establecer una llamada entre dos aplicaciones. Chequea la identidad y contexto de las entidades de aplicación, y podría aplicar criterios de autentificación.

Pasarela activa(active gateway)
: Una pasarela tratada como interfaz de red, en el sentido de que se espera que intercambie información de encaminamiento, y si no lo hace durante un periodo de tiempo, la ruta asociada con la pasarela será borrada.

Máscara de dirección(address mask)
: Una máscara de bits usada para seleccionar bits de una dirección de Internet para direccionamiento de una subred. La máscara tiene 32 bits de largo y selecciona la porción de red de la ir de Internet y uno o más bits de la porción local. A veces se la llama máscara de subred.

Resolución de direcciones(address resolution)
: Un medio para mapear direcciones del nivel de red a direcciones específicas del medio. Ver ARP.

ADMD
: Administration Management Domain. Un servicio de transporte público MHS("Message Handling System" o sistema de manejo de mensajes) X.400. Ejemplos: MCImail y ATTmail en los U.S., British Telecom Gold400mail en U.K. Los ADMDs de todos los países del mundo constituyen la troncal X.400. Ver PRMD.

Agente(agent)
: En el modelo cliente servidor, la parte del sistema que prepara e intercambia la información para una aplicación cliente o servidor. Ver NMS, DUA, MTA.

ANSI
: American National Standards Institute. El cuerpo de estandarización estadounidense. ANSI es miembro de ISO("International Organization for Standardization").

API
: Application Program Interface. Un conjunto de convenciones de llamadas que define cómo se ha de invocar un servicio a través de un software.

Capa de aplicación(application layer)
: La capa más alta en el modelo de referencia OSI que proporciona servicios de comunicación tales como E-mail y transferencia de ficheros.

Archie
: Uno de los primeros navegadores de Internet, usado para buscar ficheros en sitios FTP.

ARP
: Address Resolution Protocol. El protocolo de Internet usado para mapear dinámicamente las direcciones de Internet a direcciones físicas en redes locales. Limitado a redes que soporten broadcast por hardware.

ARPA
: Advanced Research Projects Agency. Ahora llamado DARPA, es la agencia del gobierno de U.S. que fundó ARPANET.

ARPANET
: Una red de conmutación de paquetes desarrollada a inicios de los 70'. El "abuelo" de la actual Internet. Fue relevada de sus funciones en junio de 1990.

ASN.1
: Abstract Syntax Notation One. El lenguaje OSI para describir la sintaxis abstracta. Ver BER.

Atributo(attribute)
: La forma de las piezas de información que proporciona el DS("Directory Service") X.500. La información del DS consiste en entradas, cada una conteniendo uno o más atributos. Cada atributo consiste en un identificador de tipo junto con uno o más valores. Cada operación de lectura puede realizar recuperaciones de uno o más valores.

AS(Autonomous System)
: Conjunto de "routers" que caen bajo la jurisdicción de una entidad administrativa y cooperan usando un IGP("Interior Gateway Protocol"). Ver subred.

B

Proceso en segundo plano(background process)

Un proceso que no requiere de la intervención del operador y que el ordenador puede ejecutar mientras la estación de trabajo se dedica a otra cosa.
Modo de ejecución de un programa en el que el shell no espera a la terminación del mismo antes de aceptar nuevos comandos del usuario.

Troncal(backbone)
El principal mecanismo de conectividad en un sistema distribuido jerárquico. Todos los sistemas que tengan conectividad con un sistema intermedio de la troncal tienen asegurados la conectividad entre ellos. Esto no impide que los sistemas dispongan configuraciones particulares para hacer un bypass de la troncal por razones de coste, rendimiento o seguridad.

Banda base(baseband)
Característica de cualquier tecnología de red que use una sólo portadora y que requiera que todas las estaciones conectadas a la red participen en cada transmisión. Ver banda ancha.

BER
Basic Encoding Rules. Reglas estándar para codificar unidades de datos descritas en ASN.1. A veces agrupada incorrectamente bajo el término ASN.1, que se refiere sólo a la sintaxis abstracta como lenguaje descriptivo, y no a la técnica de codificación.

BGP
Border Gateway Protocol. Un protocolo orientado a conexión(que usa TCP) desarrollado a partir de la experiencia con EGP. Ver EGP.

Big-endian
Un formato para el almacenamiento o transmisión de datos binarios en los que el bit(o byte) más significativo va primero. El convenio inverso se llama little-endian.

BITNET
Because It's Time NETwork. Una red académica basada en los sistemas mainframe de IBM interconectados por medio de líneas arrendadas de 9600 bps. BITNET acabó por fundirse con CSNET("Computer+Science Network", otra red académica) para formar CREN("Corporation for Research and Educational Networking"). Ver CSNET.

Puente(bridge)
Dispositivo que conecta dos o más redes físicas y retransmite paquetes entre ellas. Los puentes suelen construirse para filtrar paquetes, es decir, para retransmitir sólo cierto tipo de tráfico. Dispositivos relacionados son: repetidores, que simplemente transmiten señales eléctricas de un cable a otro, y "routers" totalmente funcionales que toman decisiones de encaminamiento basadas en diversos criterios.
Unidad funcional que conecta dos LANs que usan el mismo procedimiento de control lógico del enlace(LLC) pero pueden emplear diferentes procedimientos de control de acceso al medio(MAC).

Banda ancha(broadband)
Característica de cualquier red que multiplexa múltiples portadoras independientes en un sólo cable. Suele hacerse por multiplexación en frecuencia. La tecnología de banda ancha permite que diversas redes coexistan sobre un mismo cable; el tráfico de una red no interfiere con el de las demás ya que las "conversaciones" se producen a diferentes frecuencias en el "éter", de forma bastante similar al sistema de radio comercial.

Broadcast
Sistema de entrega de paquetes en el que una copia de un paquete dado se envía a todos los hosts conectados a la red. Ejemplo: Ethernet.

BSD
Berkeley Software Distribution. Término usado al describir diferentes versiones del software UNIX de Berkeley, como por ejemplo, "UNIX BSD 4.3."

C

Catenet
Una red en la que los hosts están conectados a redes con distintas características, y las redes están interconectadas mediante pasarelas("routers"). Internet es un ejemplo. Ver IONL.

CCR
Commitment, Concurrency, and Recovery. Un elemento de servicio de aplicación de OSI empleado para crear transacciones atómicas entre sistemas distribuidos. Usado principalmente para implementar operaciones indivisibles y en dos fases.

Proceso hijo(child process)
Proceso originado por un proceso padre con el que comparte recursos.

Modelo cliente/servidor(client/server model)
Una forma habitual de describir servicios de red y el modelo de usuario(programas) de esos servicios. Ejemplos pueden ser el paradigma "name-server/name-resolver" del DNS o relaciones servidor de ficheros/cliente de ficheros como NFS y hosts sin disco.

CLNP
Connectionless Network Protocol. El protocolo OSI para proporcionar el un servicio de datagramas CNS("Connectionless Network Service"). CLNP es el equivalente OSI IO, y a veces se le llama IP ISO.

CLTP
: Connectionless Transport Protocol. Suministra direccionamiento para el transporte de datos extremo-a-extremo de la conexión(por medio de un selector de transporte) y control de error(por medio de checksums), pero no garantiza la entrega ni proporciona control de flujo. El equivalente OSI de UDP.

CMIP
: Common Management Information Protocol. El protocolo de gestión de red de OSI.

CMOT
: CMIP Over TCP. Un esfuerzo para usar el protocolo de gestión de red OSI para administrar redes TCP/IP.

No orientado a conexión(connectionless)
: El modelo de interconexión en el que la comunicación se produce sin que se haya establecido una conexión previamente. A veces se le llama(imprecisamente) datagrama. Ejemplos: LANs, IP, CLNP, UDP.

Orientado a conexión(connection-oriented)
: El modelo de interconexión en el que la comunicación se produce en tres fases definidas: establecimiento de la conexión, transferencia de datos, liberación de la conexión. Ejemplos: X.25, TCP, OSI TP4, llamadas telefónicas ordinarias.

Pasarela nuclear(core gateway)
: Históricamente, una pasarela de un conjunto de ellas operado por INOC("Internet Network Operations Center") en BBN. Las pasarelas nucleares forman una parte central del encaminamiento en Internet en el sentido de que todos los grupos deben anunciar rutas a sus redes por medio de una, usando el EGP("Exterior Gateway Protocol"). Ver EGP, troncal.

COS
: Corporation for Open Systems. Un grupo de distribuidores y usuarios para testear, certificar y promocionar los productos OSI.

COSINE
: Cooperation for Open Systems Interconnection Networking in Europe. Un programa patrocinado por la EC ("European Commission"), que tiene por meta usar OSI para unir las redes de investigación europeas.

CREN
: Ver BITNET y CSNET.

CSMA/CD
: Carrier Sense Multiple Access with Collision Detection. El método de acceso que emplean las tecnologías de redes de área local como por ejemplo Ethernet.

CSNET
: Computer+Science Network. Una gran red, localizada principalmente en los U.S, pero con conexiones internacionales. Los sitios CSNET incluyen universidades, laboratorios de investigación, y algunas compañías comerciales. Ahora fusionada con BITNET para formar CREN. Ver BITNET.

D

DARPA
: Defense Advanced Research Projects Agency. La agencia el gobierno estadounidense que formó ARPANET.

Capa de enlace de datos(data link layer)
: La capa OSI responsable de la transferencia de datos a través de una sola conexión física, o series de conexiones con puentes, entre dos entidades de red.

DCA
: Defense Communications Agency. La agencia del gobierno responsable del DDN("Defense Data Network").

DCE
: Distributed Computing Environment. Arquitectura de interfaces de programación estándar y funciones de servidor para distribuir aplicaciones de forma transparente en redes de ordenadores heterogéneos. Promovida y controlada por OSF("Open Software Foundation"), un consorcio liderado por HP, DEC, e IBM. Ver ONC.

DDN
: Defense Data Network. Contiene MILNET y otras varias redes DoD.

DECnet
: Arquitectura de red de DEC("Digital Equipment Corporation").

DNS
: Domain Name System. El mecanismo distribuido de nombres/direcciones usado en Internet.

Dominio(domain)
: En Internet, una parte de una jerarquía de nombres. Sintácticamente, un nombre de dominio consiste en una secuencia de nombres(etiquetas) separados por puntos, por ejemplo, "tundra.mpk.ca.us." En OSI, "dominio" se usa generalmente como partición administrativa de un complejo sistema distribuido, como en en el PRMD ("Private Management Domain") MHS, y en DMD("Directory Management Domain").

Notación decimal con puntos(dotted decimal notation)
: La representación sintáctica para un entero de 32 bits que consiste en cuatro números de 8 bis escritos en base 10 separados por puntos. Usada para representar direcciones IP en Internet, como por ejemplo: 192.67.67.20.

DSA

Directory System Agent. El software que proporciona el DS("Directory Service" o servicio de directorio) X.500 para una parte de la DIB("Directory Information Base" o base de información de directorio). Generalmente, cada DSA es responsable de la información de directorio para una sola organización o unidad organizacional.

DUA

Directory User Agent. El software que accede al DS para el usuario del directorio, que puede ser una persona u otro elemento de software.

E

EARN

European Academic Research Network. Una red que emplea tecnología BITNET y conecta universidades y laboratorios de investigación en Europa.

EGP

Exterior Gateway Protocol. Un protocolo de encaminamiento que usan las pasarelas en redes de dos niveles. Lo usa el sistema nuclear de Internet. Ver pasarela nuclear.

Encapsulación(encapsulation)

La técnica que usan los protocolos por capas en los que una capa añade información de cabecera a la PDU de la capa inmediatamente superior. Por ejemplo, en la terminología de Internet, un paquete contendría una cabecera de la capa física, seguida de una cabecera de la capa de red, seguida de una cabecera de la capa de transporte(TCP), seguida de los datos el protocolo de aplicación.

Sistema final(end system)

Un sistema OSI que contiene procesos de aplicaciones capaces de comunicarse a través de las siete capas de protocolos de OSI. Equivale a host de Internet.

Entidad(entity)

Terminología OSI para un elemento dentro de una capa que realiza las funciones de ese nivel en un host, accediendo a la entidad inferior y proporcionando servicios a la entidad superior en los SAPs("service access points").

ES-IS

End system to Intermediate system protocol. El protocolo OSI mediante el cual los sistemas finales se anuncian a sistemas intermedios.

Ethernet

Una LAN en banda base a 10 Mbps que emplea CSMA/CD("Carrier Sense Multiple Acess with Collision Detection"). La red permite que múltiples estaciones accedan al medio a voluntad sin una coordinación previa; el mecanismo de detección de colisiones resuelve los conflictos en accesos concurrentes al medio.

EUnet

European UNIX Network.

EUUG

European UNIX Users Group.

Carácter extendido("extended character")

Un carácter diferente del tipo 7 bits. Puede ser un código de 1 byte con el octavo bit activo(128-255) o de 2 bytes(256 o mayor).

EWOS

European Workshop for Open Systems. También conocido como OIW("OSI Implementors Workshop for Europe").

F

FARNET

Federation of American Research Networks.

FDDI

Fiber Distributed Data Interface. Un estándar de redes de alta velocidad que comienza a popularizarse. El medio subyacente es la fibra óptica, y la topología es una red en anillo de doble conexión antirrotativa. Las redes FDDI suelen poderse distinguir por el "cable" de fibra naranja.

Fragmentación(fragmentation)

El proceso por el que se parte un datagrama IP en piezas más pequeñas para ajustarse a los requerimientos de una red física dada. El proceso inverso se denomina reensamblado("reassembly"). Ver MTU.

FNC

Federal Networking Council. El grupo de representantes de aquellas agencias federales involucradas en el desarrollo y uso de redes federales, especialmente de las que utilizan TCP/IP e Internet. El FNC coordina la investigación y la ingeniería. Entre los miembros actuales están representantes de DOD, DOE, DARPA, NSF, NASA y HHS.

FRICC

Federal Research Internet Coordinating Committee. Ha sido reemplazado por el FNC.

FTAM

File Transfer, Access, and Management. El servicio y protocolo de OSI de ficheros remotos.

FTP

File Transfer Protocol. El protocolo(y programa) de Internet para transferir ficheros entre hosts. Ver FTAM.

G

Pasarela(gateway)

El término original de Internet para lo que ahora se conoce como "router", o más exactamente, "router" IP. Actualmente, los términos "pasarela"("gateway") y "pasarela de aplicación"("application gateway") se refieren a sistemas que efectúan alguna traducción de un formato nativo a otro. Ejemplos son las pasarelas de correo electrónico X.400 de/a RFC 822. Ver "router".

Gopher

Herramienta de navegación por Internet que proporciona un menú de acceso a información. Muchas organizaciones la usan en vez del FTP anónimo. Ver Veronica.

GOSIP

Government OSI Profile. Una especificación del gobierno de los U.S. para los protocolos OSI.

I

IAB

Internet Activities Board. El cuerpo técnico que supervisa el desarrollo de la pila de protocolos de Internet(habitualmente, "TCP/IP"). Tiene dos fuerzas de trabajo(el IRTF y el IETF) cada una encargada de investigar en un área particular.

IANA

Internet Assigned Numbers Authority. El cuerpo técnico dentro del IAB que gestiona los estándares de protocolos en Internet. Coordina la asignación de valores a los parámetros de los protocolos.

ICMP

Internet Control Message Protocol. El protocolo empleado para gestionar errores y mensajes de control en la capa IP. Actualmente es parte del protocolo IP.

IESG

Internet Engineering Steering Group. El comité ejecutivo del IETF.

IETF

Internet Engineering Task Force. Una de las fuerzas de trabajo del IAB. El IETF es responsable de resolver las necesidades de ingeniería de Internet a corto plazo. Tiene más de 40 grupos de trabajo.

IGP

Interior Gateway Protocol. El protocolo empleado para intercambiar información de encaminamiento entre "routers" que colaboran en Internet. RIP y OSPF son ejemplos de IGPs.

IGRP

Internet Gateway Routing Protocol. El IGP que usan los "routers" de Cisco System.

Sistema intermedio(intermediate system)

Un sistema OSI que no es un sistema final, pero cuyo servicios sirven para la retransmisión de comunicaciones entre sistemas finales. Ver repetidor, puente, y "router".

Interred(internet)

Colección de redes interconectadas por un conjunto de "routers" que la permiten funcionar como una sola gran red virtual).

Internet(con "I")

La mayor interred del mundo, consistente en grandes redes troncales nacionales(como MILNET, NSFNET, y CREN) y una miriada de redes regionales y locales de campus por todo el mundo. Internet usa la pila de protocolos IP. Para estar en Internet hay que tener conectividad IP, por ejemplo, ser capaz de hacer un TELNET o un ping a otros sistemas. Las redes que sólo tienen conectividad por E-mail no se considera que pertenezcan a Internet.

Dirección de Internet(internet address)

Una dirección de 32 bits asignada a los host que usan TCP/IP. Ver notación decimal con puntos("dotted decimal notation").

IONL

Internal Organization of the Network Layer. El estándar OSI para la arquitectura detallada de la capa de red. Básicamente, particiona la capa de red en subredes interconectadas por protocolos de convergencia(equivalentes a protocolos de red), creando lo que Internet denomina "catenet".

IP

Internet Protocol. El protocolo de red de la pila de protocolos de Internet.

IP datagram

La unidad fundamental de información transmitida a través de Internet. Contiene las direcciones fuente y destino junto con datos y una serie de campos que definen la longitud del datagrama, el checksum de la cabecera y flags para indicar

cuando el datagrama ha sido(o puede ser) fragmentado.

IRTF
: Internet Research Task Force. Una de las fuerzas de trabajo del IAB. El grupo responsable de la investigación y el desarrollo de la pila de protocolos de Internet.

RDSI(ISDN)
: Integrated Services Digital Network. Una tecnología emergente que está comenzando a ser ofertado por las empresas telefónicas de todo el mundo. RDSI combina servicios de voz y digitales en un solo medio . Los estándares que definen RDSI los especifica CCITT.

IS-IS
: Intermediate system to Intermediate system protocol. El protocolo OSI que usan los sistemas intermedios para intercambiar información.

ISO
: International Organization for Standardization. Mejor conocido como el modelo de referencia OSI de siete capas. Ver OSI.

ISODE
: ISO Development Environment. Una popular implementación de las capas superiores de OSI.

J

JANET
: Joint Academic Network. Red universitaria en U.K.

JUNET
: Japan UNIX Network.

K

KA9Q
: Una popular implementación de TCP/IP y protocolos asociados para sistemas amateur de paquetes por radio.

Kermit
: Un popular programa de transferencia de ficheros y emulación de terminal.

L

Little-endian
: Un formato para el almacenamiento o la transmisión de datos binarios en los que el bit/byte menos significativo va primero. Ver big-endian.

M

ME(mail exploder)
: Parte de un sistema de entrega de correo electrónico que permite que un mensaje sea entregado en una lista de direcciones. Los MEs se emplean para implementar listas de correo. Los usuarios envían mensajes a una sola dirección(por ejemplo hacks@somehost.edu) y el SE ME encarga de distribuirlos a cada uno de los buzones de la lista de correo.

Pasarela de correo(mail gateway)
: Una máquina que conecta dos o más sistemas de correo electrónico(especialmente sistemas de correo distintos en redes diferentes) y transfiere mensajes entre ellos. A veces el mapeo y la traducción pueden ser bastante complejas, y generalmente requieren un esquema de almacenamiento-retransmisión por el que un sistema ha de recibir completamente un mensaje antes de poder realizar las traducciones pertinentes y enviarlo al siguiente sistema.

Marciano(Martian)
: Término humorístico aplicado a paquetes que aparecen de improviso en la red equivocada debido a entradas de encaminamiento corruptas. También se usa para un paquete que tiene una dirección de Internet corrupta(no registrada o mal formada).

MHS
: Message Handling System. El sistema de almacenamiento de mensajes, de agentes de usuarios y transferencia de mensajes, y de unidades de acceso que en conjunto proporcionan el correo electrónico OSI. MHS está especificado en la SR ("Series of Recommendations") CCITT X.400.

MIB

Management Information Base. Una colección de objetos que se pueden acceder a través de u protocolo de gestión de red. Ver SMI.

MILNET
: Military Network. Originalmente parte de ARPANET, MILNET se separó en 1984 para que las instalaciones militares pudieran tener un servicio de red fiable, mientras que ARPANET se siguió usando para la investigación. Ver DDN.

MIME
: Multipurpose Internet Mail Extensions. Protocolo de correo que proporciona soporte para multimedia(gráficos, audio, video) además de una compatibilidad básica con SMTP. Se describe en los RFCs 1521 y 1522. Ver SMTP.

MTA
: Message Transfer Agent. Un proceso de aplicación OSI empleado para almacenar y retransmitir mensajes en el sistema de manejo de mensajes X.400. Equivalente a un agente de correo de Internet.

MTU
: Maximum Transmission Unit. La mayor unidad de datos posible que se puede enviar sobre un medio físico dado. Ejemplo: la MTU de Ethernet es de 1500 bytes. Ver fragmentación.

Multicast
: Una forma especial de broadcast en la que las copias del paquete se entregan sólo a un subconjunto de todos los posibles destinos. Ver broadcast.

Host multi-homed(o multipuerto)
: Un ordenador conectado a más de un enlace físico de datos. Los enlaces de datos pueden estar o no conectados a la misma red.

N

Resolución de nombres(name resolution)
: El proceso de mapear un nombre a su correspondiente dirección. Ver DNS.

NetBIOS
: Network Basic Input Output System. La interfaz estándar para redes en el IBM PC y sistemas compatibles.

Dirección de red(Network Address)
: Ver dirección de Internet o dirección de red OSI.

Capa de red(network layer)
: La capa OSI responsable del encaminamiento, la conmutación y el acceso a subredes en el entorno OSI.

NFS
: Network File System. Sistema de ficheros distribuido desarrollado por Sun Microsystems que permite que un conjunto de ordenadores accedan cooperativamente a sus ficheros de forma transparente.

NIC
: Network Information Center. Originalmente sólo había uno, localizado en el SRI International, que tenía la tarea de servir a la comunidad ARPANET (y más tarde a DDN). Hoy en día, hay muchos NICs, operados por redes locales, regionales y nacionales por todo el mundo. Tales centros proporcionan asistencia al usuario, servicios de documentación, formación, y mucho más.

NIST
: National Institute of Standards and Technology(Inicialmente, NBS). Ver OIW.

NMS
: Network Management Station. El sistema responsable de gestionar una red o parte de ella. El NMS se comunica con los agentes de gestión de red, que residen en los nodos gestionados, por medio de un protocolo de gestión de red. Ver agente.

NOC
: Network Operations Center. Cualquier centro al que se le haya dado la tarea de gestionar los aspectos operacionales de una red de producción. Estas tareas incluyen la monitorización y control, resolución de problemas, asistencia al usuario, etc.

NSAP
: Network Service Access Point. El punto en el que un servicio de red OSI se hace disponible para una entidad de transporte. El NSAP se identifica con las direcciones de red OSI.

NSF
: National Science Foundation. Patrocinadores de NSFNET ("National Science Foundation Network"). Una colección de redes locales, regionales y de nivel medio en los U.S, unidas por una troncal de alta velocidad. NSFNET proporciona a los científicos acceso a una serie de superordenadores a lo largo del país.

O

OIW
: Workshop for Implementors of OSI. Llamado con frecuencia NIST OIW o NIST Workshop, es el forum regional norteamericano en el que se deciden los acuerdos sobre implementaciones OSI. Es el equivalente de EWOS en Europa

y de AOW en el Pacífico.

ONC
: Open Network Computing. Arquitectura de aplicaciones distribuidas promovida y controladas por un consorcio liderado por Sun Microsystems.

OSI
: Open Systems Interconnection. Un programa de estandarización internacional para facilitar las comunicaciones entre ordenadores de distintos fabricantes. Ver ISO.

Dirección de red OSI (OSI Network Address)
: La dirección, consistente en 20 octetos, empleada para localizar a una entidad de transporte OSI. La dirección se formatea en un IDP("Initial Domain Part") que está estandarizado para cada uno de los diversos dominios de encaminamiento, y en un DSP("Domain Specific Part"), que es responsabilidad de la autoridad de direccionamiento de ese dominio.

Dirección de presentación OSI(OSI Presentation Address)
: La dirección utilizada para localizar una entidad de aplicación OSI. Consiste en una dirección de red OSI y hasta tres selectores, cada uno para las entidades de transporte, sesión y presentación.

OSPF
: Open Shortest Path First. Propuesta de estándar para IGP. Ver IGP.

P

PCI
: Protocol Control Information. La información de protocolo que añade una entidad OSI a la SDU("service data unit") procedente de la capa superior, constituyendo en conjunto la PDU("Protocol Data Unit").

PDU
: Protocol Data Unit. Es la terminología OSI para paquete. Una PDU es un objeto de datos intercambiado por máquinas de protocolo(entidades) en una capa determinada. Las PDUs consisten tanto en la PCI("Protocol Control Information") como en los datos del usuario.

Capa física(physical layer)
: La capa OSI que proporciona los medios para activar y usar conexiones físicas para la transmisión de bits. En pocas palabras, la capa física proporciona los medios para transferir un solo bit sobre un medio físico.

Medio físico(physical media)
: Cualquier cuerpo físico que sirva de soporte a la transmisión de señales entre sistemas OSI. Se considera externo al modelo OSI, y a veces se le llama "capa 0". Se puede decir que el conector físico con el medio define la base o interfaz de nivel más bajo con la capa física, por ejemplo, la base del modelo de referencia OSI.

ping
: Packet internet groper. Un programa empleado para testear la accesibilidad de destinos enviándoles un mensaje ICMP "echo request" y esperando una respuesta.

port
: La abstracción que utilizan los protocolo de transporte de Internet para distinguir entre distintas conexiones simultáneas con el mismo host. Ver selector.

POSI
: Promoting Conference for OSI. El "gorila de 1.5 Tm" de OSI en Japón. Constituido por ejecutivos de los seis mayores fabricantes de ordenadores japoneses y de Nippon Telephone&Telegraph. Establecen políticas y movilizan recursos para promover OSI.

PPP
: Point-to-Point Protocol. Sucesor de SLIP, PPP proporciona conexiones "router"-a-"router" y host-a-red sobre circuitos tanto síncronos como asíncronos. Ver SLIP.

Dirección de presentación(Presentation Address)
: Ver dirección de presentación OSI("OSI Presentation Address").

Capa de presentación(presentation layer)
: La capa OSI que determina cómo se representa la información de la aplicación(por ejemplo, si está codificada) en su tránsito entre dos sistemas finales.

PRMD
: Private Management Domain. Un sistema de correo para organizaciones privadas basado en el sistema de manejo de mensajes X.400. Ejemplo NASAmail. Ver ADMD.

Protocolo
: Una descripción formal de los mensajes a intercambiar y de las reglas que dos o más sistemas han de seguir para intercambiar información.

Proxy
: El mecanismo por el que un sistema "da la cara" por otro sistema al responder a solicitudes de protocolo. Los sistemas proxy se emplean en la gestión de red para evitar tener que implementar pilas de protocolo enteras en dispositivos simples, tales como módems.

Proxy ARP
: La técnica con la que una máquina, habitualmente un "router", responde las solicitudes ARP dirigidas a otra máquina. Al falsificar su identidad, el "router" acepta la responsabilidad de encaminamiento paquetes a su destino real. El proxy ARP permite que un sitio use una sola dirección IP con dos redes físicas. Normalmente, el subnetting sería una solución mejor.

PSN
: Packet-Switched Node. El término moderno para nodo en ARPANET y MILNET. Solían llamarse IMPs ("Interface Message Processors"). Actualmente, los PSNs están implementados en miniordenadores BBN C30 o C300.

R

RARE
: Reseaux Associes pour la Recherche Europeenne. Asociación europea de redes de investigación.

RARP
: Reverse Address Resolution Protocol. El protocolo de Internet que usa un host sin disco para encontrar su dirección de Internet durante el arranque. RARP mapea una dirección física(hardware) a una dirección de Internet. Ver ARP.

Repetidor(repeater)
: Un dispositivo que propaga señales eléctricas de un cable a otro sin tomar decisiones de encaminamiento o sin proporcionar filtrado de paquetes. En la terminología OSI, un repetidor es la capa física de un sistema intermedio. Ver puente("bridge") y "router".

RFC
: Request For Comments. La serie de documentos, comenzada en 1969, que describe la pila de protocolos de Internet y los experimentos relacionados. No todos los RFCs describen estándares de Internet(de hecho, sólo lo hacen unos cuántos), pero todos los estándar de Internet están descritos en forma de RFCs.

RFS
: Remote File System. Un sistema de ficheros distribuido, similar al NFS, desarrollado por AT&AT y distribuido con su sistema operativo UNIX System V. Ver NFS.

RIP
: Routing Information Protocol. Un IGP("Interior Gateway Protocol") proporcionado con el UNIX de Berkeley.

RIPE
: Reseaux IP Europeens. Red TCP/IP europea continental operada por EUnet. Ver EUnet.

Rlogin
: Un servicio que ofrece el UNIX de Berkeley que permite a los usuarios de una máquina entrar en sesión en otros sistemas UNIX(para los que tengan autorización) e interactuar como si sus terminales estuvieran conectadas directamente. Similar a TELNET.

ROSE
: Remote Operations Service Element. Un RPC usado en los protocolos de aplicación OSI de manejo de mensajes, de directorio y de gestión de red.

Router
: Un sistema responsable de tomar decisiones acerca de la ruta que seguirá el tráfico de una red. Para hacerlo, utiliza un protocolo de encaminamiento con el fin de obtener información sobre la red, y algoritmos para elegir el mejor camino, basados en diversos criterios conocidos como "métricas de encaminamiento". En la terminología OSI, un "router" es la capa de red de un sistema intermedio. Ver pasarela, puente y repetidor.

RPC
: Remote Procedure Call. Un paradigma fácil y popular para implementar el modelo cliente/servidor de computación distribuida. Se envía una solicitud a un sistema remoto para que ejecute una rutina designada, con los argumentos que se le proporcionen, y los resultados se envían al llamador. Existen muchas variaciones, lo que da lugar a una serie de diferente protocolos RPC.

RTSE
: Reliable Transfer Service Element. Un servicio de aplicación OSI usado sobre redes X.25 para la negociación de las PDUs de aplicación en el SS("Session Service") y TP0. No es necesario con TP4, y no se recomienda su uso en U.S excepto al hablar con ADMDs X.400.

S

SAP
: Service Access Point. El punto en el que los servicios de una capa OSI se hacen disponibles a la capa inmediatamente superior. El SAP se denomina según la capa que proporciona los servicios en cuestión. Por ejemplo, los servicios de transporte se proporcionan en una TSAP en la cima de la capa de transporte.

Selector
: El identificador que usa una entidad OSI para distinguir entre múltiples SAPs en los que proporciona servicios a la capa inmediatamente superior. Ver puerto("port").

Capa de sesión(session layer)
: La capa OSI que proporciona los medios para el control del diálogo entre sistemas finales.

SGMP
: Simple Gateway Management Protocol. El predecesor de SNMP. Ver SNMP.

SLIP
: Serial Line IP. Un protocolo de Internet utilizado para ejecutar IP sobre líneas en serie tales como circuitos telefónicos o cables RS-232 que interconecten dos sistemas. PPP está reemplazando a SLIP. Ver PPP.

SMDS
: Switched Multimegabit Data Service. Una tecnología de red de alta velocidad en auge que las compañías telefónicas estadounidenses se disponen a ofrecer.

SMI
: Structure of Management Information. Las reglas usadas para definir los objetos que se pueden acceder por medio de un protocolo de gestión de red. Ver MIB.

SMTP
: Simple Mail Transfer Protocol. El protocolo de correo electrónico de Internet. Definido en el RFC 821, con descripciones del formato de mensajes asociados en el RFC 822.

SNA
: Systems Network Architecture. Arquitectura de red propiedad de IBM.

SNMP
: Simple Network Management Protocol. El protocolo de gestión de red electo para las interredes basadas en TCP/IP.

SPAG
: Standards Promotion and Application Group. Un grupo de fabricantes OSI europeos que establece subconjuntos de opciones y los publica en el GUS("Guide to the Use of Standards").

SQL
: Structured Query Language. El lenguaje estándar internacional para definir y acceder a bases de datos relacionales.

Máscara de subred(subnet mask)
: Ver máscara de dirección("address mask").

Subred(subnetwork)
: Un colección de sistemas finales e intermedios OSI bajo el control de un único dominio administrativo y que usan un solo protocolo de acceso a la red. Ejemplos: redes X.25 privadas, conjuntos de LANs con puentes

T

TCP
: Transmission Control Protocol. El principal protocolo de transporte de la pila de protocolos, que proporciona flujos fiables, orientados a conexión y en full-duplex. Empleado para la entrega de paquetes IP. Ver TP4.

Telnet
: El protocolo de terminal virtual en la pila de protocolos de Internet. Permite que los usuarios de un host entren en sesión en un host remoto e interactúen como usuarios normales del mismo.

three-way-handshake
: El proceso por que dos entidades de protocolo se sincronizan durante el establecimiento de la conexión.

TP0
: OSI Transport Protocol Class 0 (Simple Class). Es el protocolo de transporte más simple de OSI, útil sólo sobre una red X.25(u otra red que no pierda o dañe los datos)

TP4
: OSI Transport Protocol Class 4 ("Error Detection and Recovery Class"). El protocolo de transporte OSI más potente, útil sobre cualquier tipo de red. Es el equivalente OSI a TCP.

Transceptor(transceiver)
: Transmisor-receptor("Transmitter-receiver"). El dispositivo físico que conecta la interfaz de un host a una red de área local, tal como Ethernet. Los transceptores de Ethernet contienen una lógica que aplica señales al cable y detecta colisiones.

Capa de transporte(transport layer)
: La capa OSI responsable de la transferencia de datos fiable entre sistemas finales.

U

UA
: User Agent. Un proceso de aplicación OSI que representa un usuario humano o una organización en el sistema e manejo de mensajes X.400. crea, envía y reciba mensajes para el usuario.

UDP
: User Datagram Protocol. Un protocolo de transporte en la pila de protocolos de Internet. A al igual que TCP, se usa para

el transporte de paquetes IP, pero el intercambio de datagramas que proporciona no ofrece reconocimiento o garantía de entrega. Ver CLTP.

UUCP
: UNIX to UNIX Copy Program. Un protocolo usado para la comunicación entre sistemas UNIX.

V

Veronica
: Herramienta de búsqueda para el entorno gopher. Ver gopher.

W

WWW
: World Wide Web. Sistema global de hipertexto que soporta comunicaciones multimedia en Internet.

X

XDR
: eXternal Data Representation. Un estándar para la representación de estructuras de datos independientes de la máquina desarrollado por SUN Microsystems. Similar al ASN.1.

X/Open
: Un grupo de fabricantes de ordenadores que promueva el desarrollo de aplicaciones portables basadas en UNIX. Publican un documento llamado *X/Open Portability Guide(Guía de portabilidad de X/Open)* .

Recomendaciones X(X Recommendations)
: El documento de CCITT que describe la comunicación de datos entre estándares de red. Los bien-conocidos incluyen: estándar de conmutación de paquetes X.25, sistema de manejo de mensajes X.400, y servicios de directorio X.500.

El sistema X Window(The X Window System)
: Un popular sistema de ventanas desarrollado por el MIT e implementado en una serie de estaciones de trabajo.

 Tabla de contenidos