UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACION

ESCUELA DE INFORMATICA

"ARQUITECTURA, SERVICIOS Y APLICACIONES DEL PROTOCOLO IP"

Monografía, previa a la obtención del título:

INGENIERO EN SISTEMAS

Miembros del Tribunal de Monografías:

ING. FRANCISCO SALGADO
ING. FERNANDO BALAREZO
ING. PABLO ESQUIVEL

Autora:

Priscila del Pilar Serrano Armijos

Cuenca, Febrero 14 de 2003

Mi agradecimiento especial a
la Universidad del Azuay, a mis Profesores
que con paciencia y notable esmero
compartieron sus conocimientos,
experiencias y sobre todo su amistad.

Dedicada a mi amiga de siempre,
por su apoyo y cariño
para que realice un sueño,
que ahora compartiremos
a la distancia.

Gracias Mami.

Arquitectura, Servicios y Aplicaciones del Protocolo IP

Introducción

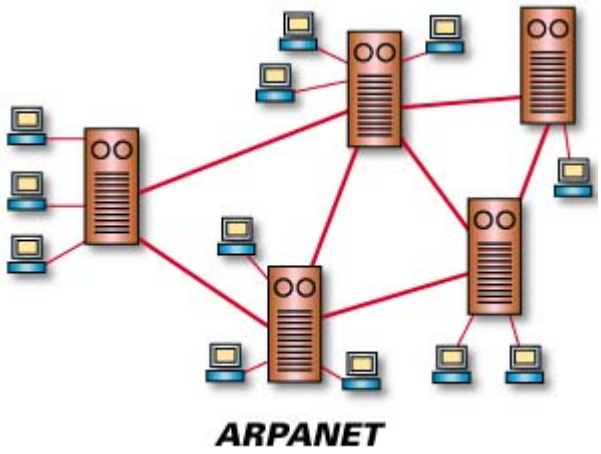
El protocolo TCP/IP, es una herramienta que permite el encaminamiento de la información a través de Internet, que se ha convertido en una parte fundamental en los sistemas de información, puesto que en la mayoría de empresas, banca, gobierno, ciencia el uso compartido de información ya sea en datos, gráficos, voz, video, etc. es tarea de todos los días, acostumbrándonos tanto a ellas que a medida que avanzamos, exigimos mejoras en los productos que manejamos, de tal forma que nuestro desempeño vaya acorde con la tecnología.

A esto se suma incluso, la situación que se produce cuando un grupo de usuarios desea comunicarse a otro grupo que no mantenga su misma tecnología, protocolos o sistema informático, y por mucho que llegaran a un acuerdo no podrían conectar físicamente sus instalaciones o aplicaciones, es entonces necesario conocer los principios básicos de la arquitectura TCP/IP de la cual parten muchas de las facilidades que mantenemos diariamente como: correo electrónico o internet y que no esta demás saber como funcionan.

En base a esta información: arquitectura, formatos, comandos, funciones, mensajes, servicios y aplicaciones podemos formar nuevas ideas de aplicación de IP o profundizar en alguno de los puntos que prometa avances o nueva información.

1. Arquitectura TCP/IP

Como resultado de la investigación y desarrollo dados en 1973, la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA), de los Estados Unidos, comenzó un programa para la investigación de tecnologías que permitieran la transmisión de paquetes de información entre redes de diferentes tipos y características. El proyecto tenía por objetivo la interconexión de redes, por lo que se le denominó "Internetting", y a la familia de redes de computadoras que surgió de esta investigación se le denominó "Internet". Los protocolos desarrollados se denominaron el Conjunto de Protocolos TCP/IP, que surgieron de dos conjuntos previamente desarrollados; los ***Protocolos de Control de Transmisión (Transmition Control Protocol) e Internet (Internet Protocol).***



TCP/IP no tiene como referencia un modelo oficial como en OSI, sin embargo se proponen cuatro capas en las que las funciones de las capas de Sesión y Presentación son responsabilidad de la capa de Aplicación y las capas de Datos y Física corresponden a la capa de acceso a la red, por ello en la arquitectura TCP-IP constan 5 niveles o capas en donde se agrupan los protocolos y se relacionan de esta forma:

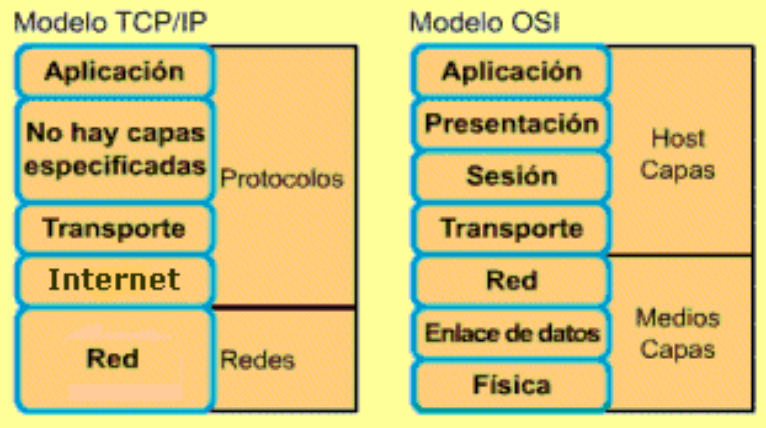
Capa de Aplicación: *Proporciona servicios que posibilitan las distintas aplicaciones a los usuarios mediante protocolos de alto nivel, por ejemplo: el protocolo de transferencia de archivos (FTP) y correo electrónico (SMTP). Protocolo de transferencia de hiper texto (http).*

Capa de Transporte: *Es la capa que garantiza la transmisión de datos de extremo a extremo, asegurándose que los datos lleguen sin errores y en la secuencia correcta. Por ejemplo: TCP (orientado a conexión) y UDP(no orientado a conexión).*

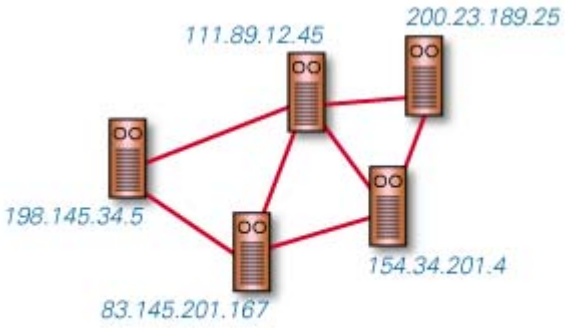
Capa de Inter-red (internet): *Este nivel permite a los hosts ingresar paquetes en cualquier red y mantenerlos viajando independiente del destino (potencialmente en una red diferente). Define un formato de paquete mediante un protocolo de inter-red conocido como IP (Internet Protocol). La función del nivel inter-red es enviar paquetes IP de un nodo a otro.*

Capa de Acceso a la Red: *Esta es respónsable del intercambio de datos entre el sistema final y al red a la cual se está conectando. El emisor debe proporcionar a la red la dirección de destino, de tal manera que ka red pueda encaminar los datos al lugar apropiado.*

Capa Física: *Es la capa que se encarga de la especificación de las características del medio de transmisión, la naturaleza de las señales, la velocidad de datos y cuestiones afines.*



2.1 Funcionamiento de TCP e IP



Dirección IP

La información en Internet viaja de host en host (ordenador conectado a una red, ya sea LAN o WAN), en pequeños paquetes, esto con el fin de ocupar mejor el espacio de las líneas de transmisión (paquetes de un mismo mensaje pueden seguir rutas diferentes, dependiendo del trafico que exista en ese momento en la red), como también para que cuando se produzca un error de transmisión, se repita solo el paquete dañado y no todo el mensaje.

TCP divide el mensaje en paquetes, cada paquete se le asigna un numero de secuencia

y la dirección de destino, estos paquetes se envían a la red, donde IP se encarga de transportarlos de host en host, hasta el destino final.

En el otro extremo TCP revisa el mensaje y comprueba que no hayan errores (dentro de la información enviada se mandan bits de paridad, para evaluar errores en la transmisión), de existir errores TCP le pide a IP que le reenvíe solo el paquete dañado, luego TCP reconstruye el mensaje original, utilizando el numero de secuencia, en resumen el protocolo IP se preocupa de transmite, transporta los distintos paquetes por las vías mas expeditas que encuentre, mientras que TCP se preocupa de la integridad y validez de los datos que se mandan y reciben.

Se necesitan dos niveles de direccionamiento, cada computador en la red debe tener una única dirección Internet que permita enviar los datos al computador adecuado, a continuación se describe paso a paso el funcionamiento del protocolo:

Un ejemplo en el que un proceso, asociado al puerto 1 en el computador A, desea enviar un mensaje a otro proceso, asociado al puerto 2 del computador B. El proceso de A pasa el mensaje al TCP con la instrucción de enviarlo al puerto 2 del conmutador B. Al IP no es necesario comunicarle la identidad del puerto destino, lo necesario es que conozca los datos que van dirigidos al computador B. Luego el IP manda el mensaje a la capa de acceso a la red con la orden de enviarlo al dispositivo de encaminamiento Z(que sería el primer punto en el camino de B).

Para controlar esta operación se debe transmitir información de control junto con los datos:

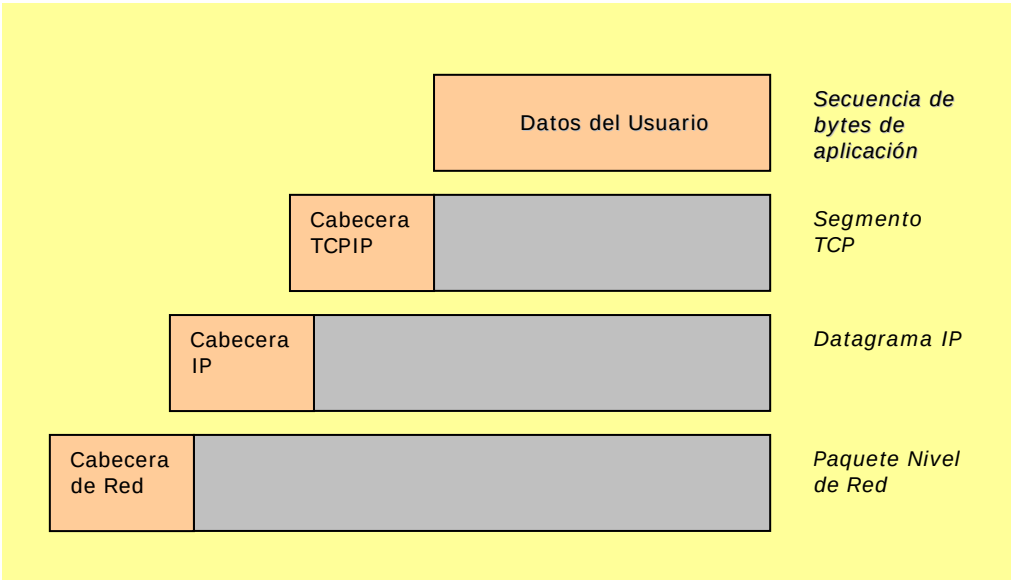
Cabecera de TCP, tiene los siguientes campos:

Puerto destino: una vez recibido el paquete, debe conocer a quién se entregarán los datos.

Número de Secuencia: Número secuencial que se envía a un puerto de destino dado.

Suma de comprobación: la entidad emisora de TCP-IP incluye un código calculado en función del resto del segmento y es comparado con el código recibido.

Luego TCP-IP pasa cada segmento IP con instrucciones para que los transmita a B, los mismos que se transmitirán a través de varias subredes y serán retransmitidos en una o varias subredes y retransmitidos a uno o varios dispositivos , solo se requiere información de control, después el IP añade la cabecera de información de control a cada segmento para formar un datagrama IP.



Unidades de Datos de Protocolo en la Arquitectura TCP/IP

En un datagrama IP se pasa a la capa de acceso de red para que se envíe a través de la primera subred, la capa de acceso a la red añade su propia cabecera, creando un paquete o trama, luego el paquete de transmite de la red al dispositivo de encaminamiento J. LA cabecera puede contener los siguientes datos:

Dirección de la Red destino, la red debe saber a que dispositivo conectado puede enviar la información.

Funciones Solicitadas, el protocolo de acceso a la red podría solicitar la utilización de ciertas funciones que ofrezca la red, como la utilización de prioridades.

En el dispositivo de encaminamiento J, se elimina la cabecera del paquete y se examina la cabecera IP. EL módulo IP del dispositivo de encaminamiento direcciona el paquete a nivel 2 hacia basándose la palabra de para que ayude con el datagrama, una cabecera de acceso a la red.

2.2 Características TCP/IP

*El TCP/IP es una **familia de protocolos** de comunicación que por su diseño permite conectar la más grande entre todas las redes existentes: **Internet**. En 1973 con ARPANET se definieron **objetivos** para la arquitectura TCP/IP:*

- *Independencia de la arquitectura del host y de la tecnología de la subred.*
- *Conectividad universal a través de la red.*
- *Confirmación de extremo a extremo.*

- *Protocolos estándares de aplicación*

Los protocolos TCP/IP presentan las siguientes ***características o metas*** *que implementan sus objetivos iniciales:*

- *Son estándares de protocolos* ***abiertos y gratuitos****. Su desarrollo y modificaciones se realizan por consenso, no a voluntad de un determinado fabricante. Cualquiera puede desarrollar productos que cumplan sus especificaciones.*
- ***Independencia a nivel software y hardware*** *Su amplio uso los hace especialmente idóneos para interconectar equipos de diferentes fabricantes, no solo a Internet sino también formando redes locales. La independencia del hardware nos permite integrar en una sola varios tipos de redes (Ethernet, Token Ring, X.25...)*
- *Proporcionan un* ***esquema común de direccionamiento*** *que permite a un dispositivo con TCP/IP localizar a cualquier otro en cualquier punto de la red.*
- *Son protocolos* ***estandarizados*** *de alto nivel que soportan servicios a los usuarios ampliamente disponibles y consistentes.*

El ***TCP y el IP son dos protocolos que pertenecen a esta colección****. Como son también los protocolos más conocidos, ha entrado en el uso común llamar TCP/IP a toda la familia. Representa una familia de protocolos, proveen a la gestión de las funciones de bajo nivel, que son necesarias para la mayoría de las aplicaciones. El TCP y el IP pertenecen a los protocolos de bajo nivel. Sobre esta base, se desarrollan otros protocolos que gestionan funciones particulares, como la transferencia de ficheros, el envío del correo electrónico, la conexión remota, el control de los usuarios que se han conectado a la red en un momento específico, con dividir impresoras y de programas aplicativos, y algo más.*

Todo esto está generalmente simplificado en un modelo cliente/servidor, en el cual el ***servidor*** *se identifica con el ordenador que proporciona un servicio específico, a través del network, y en el cual el término* ***cliente*** *se identifica con el ordenador que explota este servicio, aunque con la palabra cliente incluya también aquellos programas que uno utiliza para tener acceso a estos mismos servicios (por ejemplo navegador es el cliente típicos para tener acceso a las páginas del WWW).*

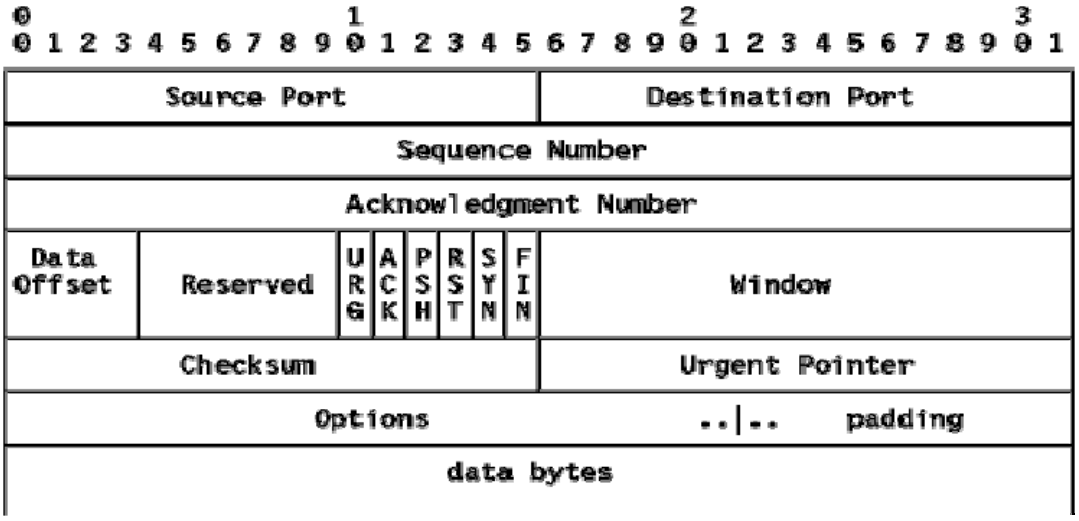
Como puede verse TCP/IP presupone ***independencia del medio físico de comunicación****, sin embargo existen estándares bien definidos a los nivel de Liga de Datos y Físico que proveen mecanismos de acceso a los diferentes medios y que en el modelo TCP/IP deben considerarse la capa de Interface de Red; siendo los más usuales el proyecto IEEE802, Ethernet, Token Ring y FDDI.*

La entrega del datagrama en IP no está garantizada porque ésta se puede retrasar, enrutar de manera incorrecta o mutilar al dividir y reensamblar los fragmentos del mensaje. Por otra parte, el IP no contiene suma de verificación para el contenido de datos del datagrama, solamente para la información del encabezado.

Es el estándar sobre el que trabaja Internet, tiene como objetivo mantener la red aunque la conexión haya sido perdida. La habilidad de conectar múltiples redes juntas y que sea capaz de sobrevivir a caídas del hardware de la subred.

Una entidad de transporte TCP acepta mensajes de longitud arbitrariamente grande procedentes de los procesos de usuario, los separa en pedazos que no excedan de 64K octetos y, transmite cada pedazo como si fuera un datagrama separado. La capa de red, no garantiza que los datagramas se entreguen apropiadamente, por lo que TCP deberá utilizar temporizadores y retransmitir los datagramas si es necesario. Los datagramas que consiguen llegar, pueden hacerlo en desorden; y dependerá de TCP el hecho de reensamblarlos en mensajes, con la secuencia correcta.

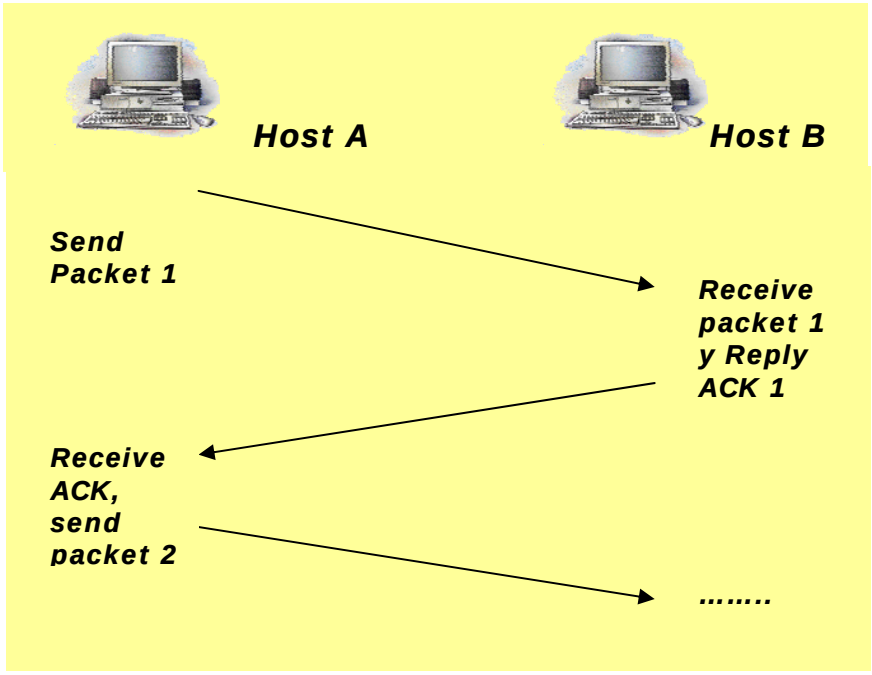
Cada octeto de datos transmitido por TCP tiene su propio número de secuencia privado. El espacio de números de secuencia tiene una extensión de 32 bits, para asegurar que los duplicados antiguos hayan desaparecidos, desde hace tiempo, en el momento en que los números de secuencia den la vuelta. TCP, sin embargo, sí se ocupa en forma explícita del problema de los duplicados retardados cuando intenta establecer una conexión, utilizando el protocolo de ida-vuelta-ida para este propósito. En la figura se muestra la cabecera que se utiliza en TCP. La primera cosa que llama la atención es que la cabecera mínima de TCP sea de 20 octetos. A diferencia de la clase 4 del modelo OSI, con la cual se puede comparar a grandes rasgos, TCP sólo tiene un formato de cabecera de TPDU(llamadas mensajes). Enseguida se analizará minuciosamente campo por campo, esta gran cabecera. Los campos Puerto fuente y Puerto destino identifican los puntos terminales de la conexión(las direcciones TSAP de acuerdo con la terminología del modelo OSI). Cada hostal deberá decidir por sí mismo cómo asignar sus puertos.



Los campos Número de secuencia y Asentimiento en superposición efectúan sus funciones usuales. Estos tienen una longitud de 32 bits, debido a que cada octeto de datos está numerado en TCP.

La Longitud de la cabecera TCP indica el número de palabra de 32 bits que están contenidas en la cabecera de TCP. Esta información es necesaria porque el campo Opciones tiene una longitud variable, y por lo tanto la cabecera también.

Después aparecen seis banderas de 1 bit. Si el Puntero acelerado se está utilizando, entonces URG se coloca a 1. El puntero acelerado se emplea para indicar un desplazamiento en octetos a partir del número de secuencia actual en el que se encuentran datos acelerados. Esta facilidad se brinda en lugar de los mensajes de interrupción. El bit SYN se utiliza para el establecimiento de conexiones. La solicitud de conexión tiene SYN=1 y ACK=0, para indicar que el campo de asentimiento en superposición no se está utilizando. La respuesta a la solicitud de conexión si lleva un asentimiento, por lo que tiene SYN=1 y ACK=1. En esencia, el bit SYN se utiliza para denotar las TPDU CONNECTION REQUEST Y CONNECTION CONFIRM, con el bit ACK utilizado para distinguir entre estas dos posibilidades. El bit FIN se utiliza para liberar la conexión; especifica que el emisor ya no tiene más datos. Después de cerrar una conexión, un proceso puede seguir recibiendo datos indefinidamente. El bit RST se utiliza para reiniciar una conexión que se ha vuelto confusa debido a SYN duplicados y retardados, o a caída de los hostales. El bit EOM indica el Fin del Mensaje.



El control de flujo en TCP se trata mediante el uso de una ventana deslizante de tamaño variable. Es necesario tener un campo de 16 bits, porque la ventana indica el número de octetos que se pueden transmitir más allá del octeto asentido por el campo ventana y no cuántas TPDU.

El código de redundancia también se brinda como un factor de seguridad extrema. El algoritmo de código de redundancia consiste en sumar simplemente todos los datos,

considerados como palabras de 16 bits, y después tomar el complemento a 1 de la suma.

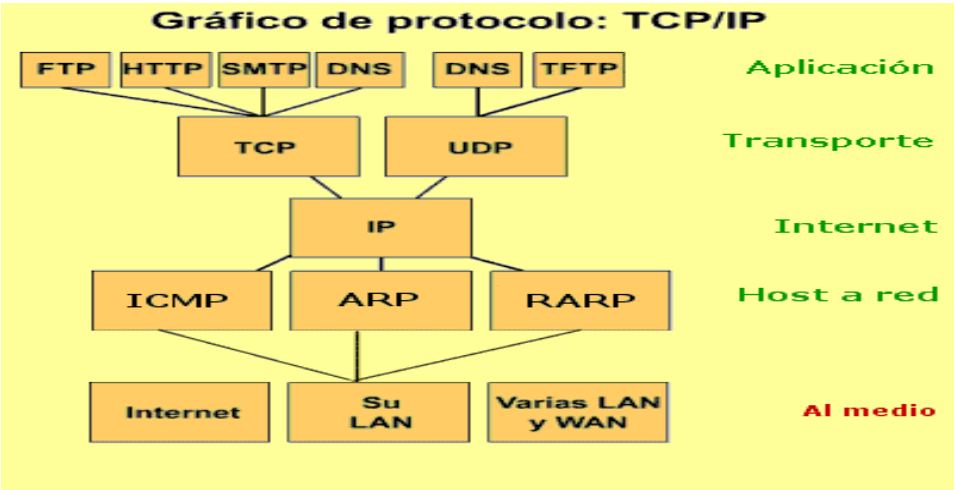
El campo de Opciones se utiliza para diferentes cosas, por ejemplo para comunicar tamaño de tampones durante el procedimiento de establecimiento.

Por otro lado, el modelo TCP/IP, puede presentar algunos problemas que se deben tomar en cuenta:

- *Este modelo no distingue con claridad los conceptos de servicio, interfaz y protocolo. En el modelo TCP/IP no se hace una diferenciación entre las especificaciones y la implementación y por lo tanto no es un buen ejemplo para construir redes nuevas.*
- *Otra limitación del modelo es que no es general y no se puede utilizar para describir otra pila de protocolos.*
- *La capa de acceso a la red no es una capa en el sentido estricto sino que es un interface entre la red y la capa de enlace de datos.*
- *En este modelo TCP/IP no se distingue entre la capa física (que tiene que ver con las características de transmisión en el alambre de cobre, la fibra óptica y la comunicación inalámbrica) y la capa de enlace de datos (encargada de delimitar el inicio y el fin de los marcos y transferirlos de un lado a otro con el grado deseado de confiabilidad) cuando en realidad son completamente diferentes y por lo tanto ambas capas deberían aparecer separadas en un modelo bien diseñado.*
- *Aunque los protocolos IP y TCP se diseñaron cuidadosamente, las implementaciones de los otros se distribuían gratuitamente, se utilizan mucho y por lo tanto eran difíciles reemplazar aunque tuvieran problemas de diseño.*

2.3 Aplicaciones TCP/IP

Los protocolos más importantes de TCP/IP, están identificados en el gráfico:



*En la capa de **Aplicación**, aparecen distintas tareas de red que probablemente se usan casi todos los días, las aplicaciones son controladas por algunos de estos protocolos:*

***FTP**: File Transfer Protocol (Protocolo de transporte de archivos). Se utiliza para enviar ficheros de texto o binarios, de un sistema a otro bajo el control del usuario, por medio de su identificador y contraseña, así como también las acciones que se darán sobre el fichero. Una vez que el fichero se especificó y la transferencia es aceptada, se establece una segunda conexión TCP, en la que se materializa la transferencia, sin necesidad de transmitir información extra o cabeceras generadas por la capa de aplicación. Una vez finalizada la transferencia, se usa la conexión de control para especificar el fin, y la misma conexión estará disponible para próximas transferencias.*

***TELNET**: protocolo de servicio de conexión remota (remote login). Es un emulador de terminal que permite acceder a los recursos y ejecutar programas en un ordenador remoto; es decir, nos permite conectarnos a un equipo remoto y actuar sobre él como si estuviéramos físicamente conectados al mismo. De esta manera se puede abrir una sesión (entrar y ejecutar comandos) o acceder a otros servicios especiales: como por ejemplo consultar un catálogo de una biblioteca para buscar un libro, leer un periódico electrónico, buscar información sobre una persona, etc.*

La comunicación entre cliente y servidor se maneja con órdenes internas, que no son accesibles por los usuarios. Todas las órdenes internas de Telnet consisten en secuencias de 2 ó 3 bytes, dependiendo del tipo de orden. Los problemas más frecuentes que suelen darse con Telnet son del tipo de la configuración de la terminal. En principio, cada computadora acepta que las terminales que se conectan a ella sean de algún tipo determinado (normalmente VT100 o VT200) y si nuestro software de Telnet no es capaz de emular estos tipos de terminales lo suficientemente bien, pueden aparecer caracteres extraños en la pantalla o que no consigamos escribir con nuestro teclado un determinado carácter. La mayoría de las implementaciones de Telnet no proporciona capacidades gráficas.

***HTTP**: Hypertext Transfer protocol (Protocolo de transferencia de hipertexto). Proporciona el servicio de páginas web, mediante el cual podemos solicitar éstas a un servidor web y visualizarlas en los navegadores clientes.*

***SMTP**: Simple Mail transport protocol (Protocolo de transporte de correo simple). Proporciona el servicio de correo electrónico, permitiendo enviar mensajes a otros usuarios de la red. Estos mensajes se envían primero a unos equipos servidores especiales, desde los cuales pueden ser descargados por el destinatario final.*

***DNS**: Domain Name Service (Servicio de nombre de dominio). Proporciona el servicio de traducción de nombres de domino en direcciones IP reales.*

***TFTP**:Trival File transport protocol (Protocolo de transporte de archivo trivial). El modelo TCP/IP enfatiza la máxima flexibilidad, en la capa de aplicación, para los diseñadores de software. Es un protocolo extremadamente simple para transferir ficheros. Está implementado sobre UDP y carece de la mayoría de las características de FTP. La única cosa que puede hacer es leer/escribir un fichero de/a un servidor. No*

tiene medios para autentificar usuarios: es un protocolo inseguro. Cualquier transferencia comienza con una petición de lectura o escritura de un fichero. Si el servidor concede la petición, la conexión se abre y el fichero se envía en bloques de 512 bytes (longitud fija). Los bloques del fichero están numerados consecutivamente, comenzando en 1. Un paquete de reconocimiento debe reconocer cada paquete de datos antes de que el próximo se pueda enviar. Se asume la terminación de la transferencia cuando un paquete de datos tiene menos de 512 bytes. Casi todos los errores causarán la terminación de la conexión (por falta de fiabilidad). Si un paquete se pierde en la red, ocurrirá un timeout, después de que la retransmisión del último paquete (datos o reconocimiento) tuviera lugar.

En la capa de ***Transporte,*** *para regular el flujo de información, garantizar la conectividad de extremo a extremo aparecen dos protocolos principales:*

Protocolo de Control de Transmisión *(TCP, que es un protocolo orientado a la conexión que permite a un flujo de bytes (mensaje) originado en una máquina sea enviado sin errores hasta cualquier otra máquina en la inter-red. Fragmenta mensajes grandes en paquetes. En el destino el proceso TCP del que recibe, reensambla los mensajes a partir de los paquetes recibidos. TCP también maneja control de flujo para asegurar un rápido envío y sincronizar las máquinas rápidas con las máquinas lentas y viceversa.*

Protocolo de Datagrama de Usuario (UDP)*, es un protocolo para aplicaciones que no quieren la secuencia del TCP ó control de flujo y desean construir uno propio. Son ampliamente usados por aplicaciones cliente-servidor, pero los servicios no son confiables, ya que la información puede perderse en el camino y no hay mecanismo de seguimiento ni control para su recuperación.*

En la capa de ***Internet o red*** *existe solamente un protocolo, el* ***protocolo Internet (IP)****, independientemente de la aplicación que solicita servicios de red o del protocolo de transporte que se utiliza. IP sirve como protocolo universal que permite que cualquier computador en cualquier parte del mundo pueda comunicarse en cualquier momento, y es la base fundamental de Internet. El protocolo IP define las unidades de transferencia de datos, denominadas paquetes o datagramas, y se encarga de su transferencia desde el host origen al host destino. Se implementa por software.*

El papel de la capa IP es averiguar cómo encaminar paquetes o datagramas a su destino final, lo que consigue mediante el protocolo IP. Para hacerlo posible, cada interfaz en la red necesita una ***dirección IP****. Una dirección IP identifica un host de forma única (más datos en el siguiente capítulo). Dos host no pueden tener una misma direción IP pública, pero si pueden tener la misma IP si pertenecen a dos redes privadas diferentes.*

A pesar de ser el protocolo IP el único encargado del direccionamiento a nivel general, a nivel interno existe otro protocolo ámpliamente usado, el ***RIP (Protocolo de Informaciónde Ruteo)****, conocido también por el programa que lo implementa, el Route Daemon. Es consecuencia directa de la impementación del ruteo vectro-distancia en redes locales, y divide las máquinas participantes en el proceso de ruteo en activas y pasivas. Los routers activos anuncian sus rutas a los otros difundiendo un*

mensaje cada 30 segundos, mensaje que contiene información tomada de la base de datos de ruteo actualizada. Las máquinas pasivas listan y actualizan sus rutas en base a estos mensajes.

En la capa de ***Acceso a la Red:*** *TCP/IP no especifica claramente un protocolo de nivel de enlace de datos, son necesarios mecanismos para traducir las direcciones IP a direcciones que entendieran el software de capa de enlace de datos por sobre el que corre TCP/IP y para controlar posibles errores a nivel de subred. Por eso se introdujeron protocolos específicos, como:*

ICMP *(Protocolo de Mensajes de Control y Error de Internet): es de carcaterísticas similares a UDP, pero con un formato mucho más simple, y su utilidad no está en el transporte de datos de usuario, si no en controlar si un paquete no puede alcanzar su destino, si su vida ha expirado, si el encabezamiento lleva un valor no permitido, si es un paquete de eco o respuesta, etc. Es decir, se usa para manejar mensajes de error y de control necesarios para los sistemas de la red, informando con ellos a la fuente original para que evite o corrija el problema detectado. ICMP proporciona así una comunicación entre el software IP de una máquina y el mismo software en otra.*

ARP *(Protocolo de Resolución de Direcciones): una vez que un paquete llega a una red local mediante el ruteo IP, la entrega del mismo al host destino se debe realizar forzosamente mediante la dirección MAC del mismo (número de la tarjeta de red), por lo que hace falta algún mecanismo capaz de transformar la dirección IP que figura como destino en el paquete en la dirección MAC equivalente, es decir, de obtener la relación dirección lógica-dirección física. Esto sucede así porque las direcciones Ethernet y las direcciones IP son dos números distintos que no guardan ninguna relación entre ellos.*

De esta labor se encarga el protocolo ARP, que en las LAN equipara direcciones IP con direcciones Ethernet (de 48 bits) de forma dinámica, evitando así el uso de tablas de conversión. Mediante este protocolo una máquina determinada (generalmente un router de entrada a la red o un swicht) puede hacer un broadcast mandando un mensaje, denominado petición ARP, a todas las demás máquinas de su red para preguntar qué dirección local pertenece a alguna dirección IP, siendo respondido por la máquina buscada mediante un mensaje de respuesta ARP, en el que le envía su dirección Ethernet. Una vez que la máquina peticionaria tiene este dato envía los paquetes al host destino usando la direción física obtenida.

RARP *(ARP por Réplica): permite que una máquina que acaba de arrancar o sin disco pueda encontrar su dirección IP desde un servidor. Para ello utiliza el direccionamiento físico de red, proporcionando la dirección hardware física (MAC) de la máquina de destino para identificar de manera única el procesador, transmitiendo por difusión la solicitud RARP. Una vez que la máquina obtiene su dirección IP la guarda en memoria, y no vuelve e usar RARP hasta que no se inicia de nuevo.*

2. Servicios IP

Los servicios que se proporcionan entre las capas de protocolos se definen como primitivas y parámetros. La primitiva especifica la función que se va a ofrecer y los parámetros que se utilizan para pasar información de control.

El protocolo IP utiliza dos primitivas de servicio en la interfaz con la siguiente capa superior. Por ejemplo: Se usa la primitiva "envío" para solicitar transmisión de datos y la primitiva "entrega" utiliza IP para notificar al usuario la llegada de los datos, los parámetros asociados con estas primitivas son:

- *Dirección de origen, dirección de red que envía los datos.*
- *Dirección destino, dirección de red de destino.*
- *Protocolo, entidad de protocolo recipiente.*
- *Indicadores del tipo de servicio, especifica el tratamiento de los datos al transmitirlos por los componentes de la red.*
- *Identificador, es la unión de la dirección origen, destino y protocolo para identificar los datos.*
- *Identificador de no fragmentación, indica si IP puede fragmentarse para el transporte.*
- *Tiempo de vida, medida dada en segundos.*
- *Datos de opción, opciones solicitadas por IP.*
- *Datos, información a transmitir.*

Los campos identificador, identificador de no fragmentación y tiempo de vida*, dan instrucciones a IP, pero no le interesan al usuario de "entrega". Al hacer el envío, el* tipo servicio*, es para solicitar un servicio especial, que puede ayudar a definir las opciones de encaminamiento, como por ejemplo:*

Precedencia, Dispone de ocho niveles de precedencia, que pueden ayudar a darle cierta prioridad al datagrama.
Seguridad, Puede ser Normal o Alto, este último indica una petición más para evitar la pérdida o daño del datagrama.
Retardo, Puede ser Normal o bajo, este último indica una petición más para evitar retardo en el datagrama.
Rendimiento, Puede ser Normal o Alto, este último indica una petición para maximizar el rendimiento del datagrama.

Dentro de los parámetros existen opciones que permiten futuros aplicaciones, como por ejemplo:

- *Seguridad, agrega una etiqueta de seguridad en el datagrama.*
- *Encaminamiento por la fuente, elabora una lista secuencial de direcciones de dispositivos que especifican la ruta a seguir.*
- *Registro de la Ruta, en un campo guarda la secuencia de encaminamiento seguidos por el datagrama.*
- *Identificación de secuencia, identifica los recursos reservados utilizados para un servicio de secuencia.*

- *Marcas de tiempo, una marca temporal a los datos según su paso por ellos, dada en la entidad origen o en todos los dispositivos de encaminamiento.*

2.1 Protocolo IP

El protcolo IP(Internet Protocol) está basado en la idea de datagramas, los cuales son transportados transparentemente, pero no siempre con seguridad, desde la fuente hasta el destinatario, con la posibilidad de recorrer varias redes hasta llegar a su destino final..

El protocolo IP no está orientado a conexión y no es confiable, ya que manda paquetes (datagramas) sin contar con mecanismos de verificación de entrega y sin comprobación de errores. Aunque se debe tomar en cuenta que el protocolo superior, TCP, se encarga de corregir estas debilidades. En cuanto al ruteo o direccionamiento de los datagramas, se puede realizar paso a paso por todos los nodos o mediante tablas de rutas estáticas o dinámicas.

Este protocolo es usado por los de la capa de transporte para encaminar los datos a su destino, siendo ésta su misión última, por lo que no se preocupa de la integridad de la información que contienen los paquetes.

Para poder direccionar los datagramas, IP introduce una nueva cabecera en los mismos, armada por 32 bits, y que contiene diferentes datos necesarios para poder enrutar los paquetes, como:

- ***Ver(4 bits)**: Versión de IP que se emplea para construir el Datagrama. Se requiere para que quien lo reciba lo interprete correctamente. La actual versión IP es la 4.*
- ***Longitud de la Cabecera Internet (Hlen, 4 bits) :** Tamaño de la cabecera en palabras de 32 bits, lo mínimo es de 5, que corresponde a una cabecera de 20 octetos.*

- ***Tipo de Servicio (TOS, 8 bits):*** *La gran mayoría de los Host y Routers ignoran este campo. Su estructura es:*

Prioridad	D	T	R	Sin Uso

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

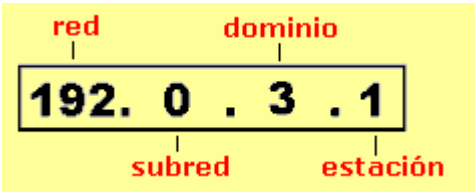
- ***Longitud Total(16 bits)****: Mide en bytes la longitud de todo el Datagrama. Permite calcular el tamaño del campo de datos: Datos = Longitud Total – 4 * Hlen.*
- ***Identificación(16 bits)****: identifica al Datagrama, permite implementar números de secuencias y reconocer los diferentes fragmentos de un mismo Datagrama, pues todos ellos comparten este numero(dirección origen, destino y protocolo).*
- ***Banderas(Flags, 3 bits):*** *el primer campo está reservado. El segundo, llamado bit de No - Fragmentación significa que: si es 0, puede fragmentarse el Datagrama o si es 1, no puede fragmentarse el Datagrama. El tercer bit es llamado Más – Fragmentos y significa: 0 = Unico fragmento o Ultimo fragmento, 1 = aun hay más fragmentos. Cuando hay un 0 en más – fragmentos, debe evaluarse el campo desp. De Fragmento: si este es cero, el Datagrama no esta fragmentado, si es diferente de cero, el Datagrama es un ultimo fragmento.*
- ***Desp. De Fragmento(13 bits)****: A un trozo de datos se le llama Bloque de Fragmento. Este campo indica el tamaño del desplazamiento en bloques de fragmento con respecto al Datagrama original, empezando por el cero.*
- ***TTL(8 bits)****: Tiempo de Vida del Datagrama, especifica el numero de segundos que se permite al Datagrama circular por la red antes de ser descartado.*
- ***Protocolo****: Especifica que protocolo de alto nivel se empleó para construir el mensaje transportado en el campo datos de Datagrama IP. Algunos valores posibles son: 1 = ICMP, 6 = TCP, 17 = UDP, 88 = IGRP (Protocolo de Enrutamiento de Pasarela Interior de CISCO).*
- ***Checksum(16 bits)****: Es un campo que se calcula haciendo el complemento a uno de cada palabra de 16 bits del encabezado, sumándolas y haciendo su complemento a uno. Esta suma hay que recalcularla en cada nodo intermedio debido a cambios en el TTL o por fragmentación.*
- ***Dirección IP de la Fuente(32 bits)****:identifica la red y el sistema final conectado a la red especificada.*
- ***Dirección IP del Destino(32 bits)*** *identifica la red y el sistema final conectado a la red especificada.*
- ***Opciones IP(variable)****: Existen hasta 40 bytes extra en la cabecera del Datagrama IP que pueden llevar una o más opciones. Su uso es bastante raro.*
    - *Uso de Ruta Estricta (Camino Obligatorio)*
    - *Ruta de Origen Desconectada (Nodos Obligatorios)*

- *Crear registro de Ruta*
- *Marcas de Tiempo*
- *Seguridad Básica del Departamento de Defensa*
- *Seguridad Extendida del Departamento de Defensa*

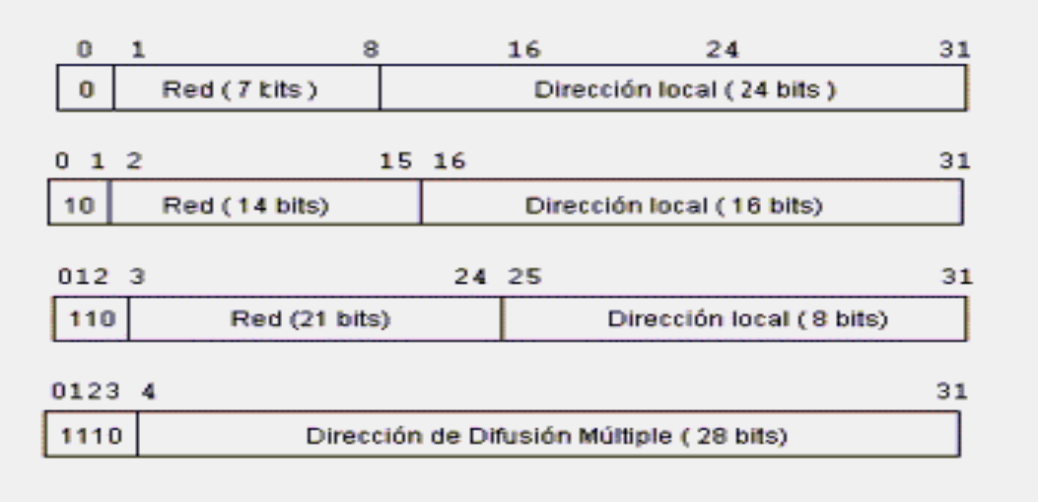
- ***Relleno (variable):*** *se usa para asegurar que la cabecera del datagrama tiene una longitud múltiplo de 32 bits.*
- ***Datos (variable)****: este campo debe tener una longitud múltiplo de 8 bits.*

2.2 Direcciones IP (clases, subredes, máscaras de subred)

En cada ordenador que se encuentre conectado a una red esta identificada su ***dirección IP****. Esta dirección es un número de 32 bit que debe ser único para cada host, y normalmente suele representarse como cuatro cifras de 8 bit separadas por puntos que identifica tanto un ordenador concreto como la red a la que éste pertenece, ya que el sistema de direcciones IP es un sistema jerárquico.*



Al tomar en cuenta que en Internet se encuentran conectadas redes de tamaños muy diversos, se establecieron cuatro formatos diferentes de direcciones, que se usan de acuerdo al tamaño de la red, aunque últimamente se ha añadido la Clase E para un futuro) aparecen en la figura:



Conceptualmente, cada dirección está compuesta por un par (RED (netid), y Dir. Local (hostid)) en donde se identifica la red y el host dentro de la red.

La clase se identifica mediante las primeras secuencias de bits, a partir de los 3 primeros bits (de orden más alto).

*Las direcciones de **Clase A** corresponden a redes grandes con muchas máquinas. Las direcciones en decimal estan entre 1 a 126 (lo que permite hasta 1.6 millones de hosts por que .estas direcciones utilizan únicamente este primer byte para identificar la red, quedando los otros tres bytes disponibles para cada uno de los hosts que pertenezcan a esta misma red). Este tipo de direcciones es usado por redes muy extensas, son pocas las organizaciones que obtienen una dirección de "clase A". Lo normal para las grandes organizaciones es que utilicen una o varias redes de "clase B".*

*Las direcciones de **Clase B** sirven para redes de tamaño intermedio, y el rango de direcciones varía desde el 128 hasta el 191. Esto permite tener 16320 redes con 65024 host en cada una. El identificador de la red se obtiene de los dos primeros bytes de la dirección, teniendo que ser un valor entre 128.1 y 191.254 (no es posible utilizar los valores 0 y 255 por tener un significado especial). Los dos últimos bytes de la dirección constituyen el identificador del host permitiendo, por consiguiente, un número máximo de 64516 ordenadores en la misma red. En caso de que el número de ordenadores que se necesita conectar fuese mayor en las grandes organizaciones, sería posible obtener más de una dirección de "clase B".*

*Las direcciones de **Clase C** tienen sólo 8 bits para la dirección local o de anfitrión (host) y 21 bits para red. Las direcciones de esta clase están comprendidas entre 192 y 223, lo que permite cerca de 2 millones de redes con 254 hosts cada una. Estas direcciones permiten un menor número de host que las anteriores, aunque son las más numerosas pudiendo existir un gran número redes de este tipo (más de dos millones).*

Tabla de direcciones IP de Internet.

Clase	Primer byte	Identificación de red	Identificación de hosts	Número de redes	Número de hosts
A	1 .. 126	1 byte	3 byte	126	16.387.064
B	128 .. 191	2 byte	2 byte	16.256	64.516
C	192 .. 223	3 byte	1 byte	2.064.512	254

*Las direcciones de **Clase D** se usan con fines de multidifusión, cuando se quiere una difusión general a más de un dispositivo. El rango es desde 224 hasta 239, esto quiere decir que, las direcciones de clase E (aunque su utilización será futura) comprenden el rango desde 240.0.0.0 hasta el 247.255.255.255.*

*En la clasificación de direcciones hay ciertos números que no se usan, por que están se encuentran **reservados** para un posible uso futuro, como por ejemploen las que el primer byte sea superior a 223 (clases D y E), mientras que el valor 127 en el primer*

byte se utiliza en algunos sistemas para propósitos especiales. Cabe recalcar que los valores 0 y 255 en cualquier byte de la dirección no pueden usarse normalmente por tener otros propósitos específicos.

El número 0 está reservado para las máquinas que no conocen su dirección, pudiendo utilizarse tanto en la identificación de red para máquinas que aún no conocen el número de red a la que se encuentran conectadas, en la identificación de host para máquinas que aún no conocen su número de host dentro de la red, o en ambos casos.

CLASE	RANGO DE DIRECCIONES IP RESERVADAS
A	10.x.x.x
B	172.16.x.x - 172.31.x.x
C	192.168.0.x - 192.168.255.x

El número 255 se reserva para el broadcast, que es necesario cuando se envía un mensaje para todos los sistemas conectados a la misma red. Esto puede ser útil si se necesita enviar el mismo datagrama a un número determinado de sistemas, siendo más eficiente que enviar la misma información a cada ordenador. Otra situación para el uso de broadcast es cuando se quiere convertir el nombre por dominio de un ordenador a su correspondiente número IP y no se conoce la dirección del servidor de nombres de dominio más cercano.

Cuando se requiere el uso del broadcast se utiliza una dirección compuesta por el identificador normal de la red y por el número 255 (todo unos en binario) en cada byte que identifique al host. Sin embargo, por conveniencia también se permite el uso del número 255.255.255.255 con la misma finalidad, de forma que resulte más simple referirse a todos los sistemas de la red.

El broadcast es una característica que se encuentra implementada de formas diferentes dependiendo del medio utilizado, y por lo tanto, no siempre se encuentra disponible. En ARPAnet y en las líneas punto a punto no es posible enviar broadcast, pero sí que es posible hacerlo en las redes Ethernet, donde se supone que todos los ordenadores prestarán atención a este tipo de mensajes.

Aparte de las IPs reservadas, existen otras ***direcciones especiales*** *que tienen un significado especial y que no se pueden asignar a ningún host de una red. Si nuestro host pertenece por ejemplo una red de clase C, de rango de direcciones IP 220.2.36.x, las siguientes direcciones son especiales:*

220.2.36.0.....dirección propia de la red

220.2.36.255.....dirección de broadcast de la red 220.2.36.0

255.255.255.255.....dirección de broadcast de nuestra red

0.0.0.0.....nuestra propio host

127.0.0.x.....loopback de nuestro propio host

0.0.0.25.....host 25 de nuestra propia red

Un mensaje broadcast a la red de clase B 140.26.5.95 se haría mediante el IP 140.26.255.255, así el mensaje llegaría a todos los host de esa red. Existe además la dirección de ***loopback*** *(generalmente la 127.0.0.1) corresponde a nuestro propio host, y se utiliza para acceder a los servicios TCP/IP del mismo, así por ejemplo: un servidor web local y se requiere acceder a las páginas del mismo vía HTTP, para ello en la barra de direcciones se ingresa la dirección 127.0.0.1 y el puerto en el que está escuchando el servidor es el 80, la dirección de acceso sería la 127.0.0.1:8080.*

Las direcciones IP, pueden clasificarse además, por su accesibilidad o por su perdurabilidad.

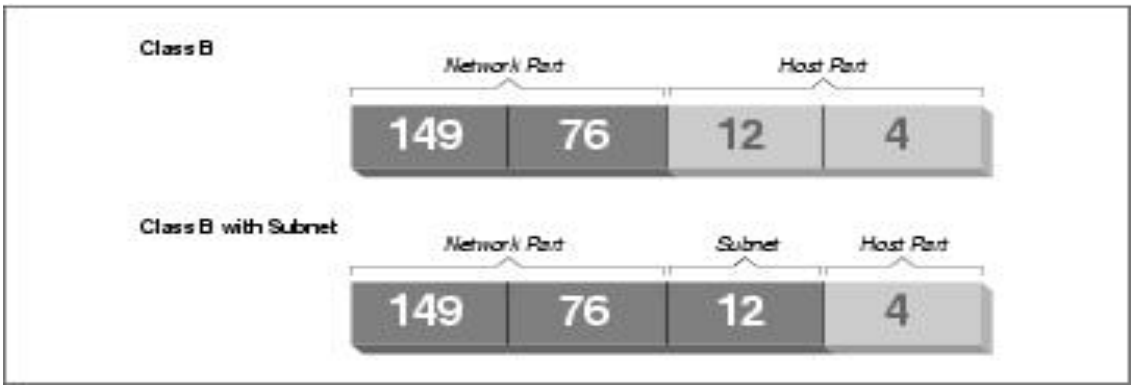
En el primer caso, pueden ser Direcciones IP Pública, siempre visibles a todos los host conectados a Internet, o las Direcciones IP Privadas, que son solo visibles a los host de su propia red, y en el segundo caso, por perdurabilidad, existen las direcciones IP Estáticas, que son asignadas en forma fija y siempre será la misma, y las direcciones Ip Dinámicas, que son asignadas por el host al momento de conectarse.

Subredes y Máscaras de Subred

La definición de subred, nace de la necesidad de dividir una red en subredes, puesto que el rendimiento de una red se ve afectado, por ejemplo a medida que aumenta el número de host, aumentarán también el número de transmisiones de broadcast, pudiendo con este tráfico congestionar toda la red, por consumo excesivo del ancho de banda, ya que todos los host están enviando muchas peticiones de ARP, RIP, DNS, etc.

Al dividir la red primaria en una serie de subredes, van a funcionar a nivel de envío y recepción de paquetes, como una red individual, aunque todas pertenezcan a la misma red principal y al mismo dominio. De esta forma, aunque la red en su conjunto tendrá una dirección IP única, administrativamente, podremos considerar subredes bien diferenciadas, consiguiendo con ello un control del tráfico de la red y una limitación de las peticiones de broadcast que la atraviesan.

En el gráfico se muestra como 149.76.12.4, se interpreta de forma distinta cuando la dirección viene dada como una red de clase B ordinaria y cuando se usa como subred.



División de una red de clase B en subredes

Generar subredes es únicamente una división interna de la red. Las subredes se generan por el propietario de la red (o el administrador). Frecuentemente se crean para reflejar límites determinados, como físicos (entre dos Ethernets), administrativos (entre dos departamentos), o geográficos (entre dos ubicaciones distintas), y la autoridad de cada subred se delega a alguna persona de contacto, pues la estructura afecta solo al funcionamiento interno de la red y es completamente invisible para el mundo exterior.

Una subred se responsabiliza de enviar datagramas a un cierto rango de direcciones IP, esto es una extensión del concepto de dividir campos de bits, como en las clases A, B, y C. De cualquier forma, la parte de red se extiende ahora para incluir algunos bits de la parte del puesto. El número de bits que se interpreta como el número de subred viene dado por la llamada ***máscara de subred*** *o máscara de red. Este es también un número de 32 bits, que especifica la máscara de bit para la parte de red de la dirección IP.*

Cuando dos o más redes diferentes se encuentran conectadas entre sí por medio de un router, éste debe disponer de algún medio para diferenciar los paquetes que van dirigidos a los host de cada una de las redes. Es aquí donde entra el concepto de máscara de red, que es una especie de dirección IP especial que permite efectuar este enrutamiento interno de paquetes.

Dada una dirección IP de red cualquiera, la máscara de red asociada es aquella que en binario tiene todos los bits que definen la red puestos a 1 (255 en decimal), y los bits correspondientes a los host puestos a 0 (0 en decimal). Así, las máscaras de red de los diferentes tipos de rsdes son:

Red Clase A.....Máscara de red=255.0.0.0

Red Clase B.....Máscara de red=255.255.0.0

Red Clase C.....Máscara de red=255.255.255.0

La máscara de red posee la importante propiedad de que cuando se combina con la dirección IP de un host se obtiene la dirección propia de la red en la que se encuentra el mismo. Cuando al router que conecta varias redes le llega un paquete saca de él la dirección IP del host destino y realiza una operación AND lógica entre ésta IP y las diferentes máscaras de red de las redes que une, comprobando si el resultado coincide con alguna de las direcciones propias de red. Este proceso de identificación de la red destino de un paquete (y del host al que va dirigido el paquete) se denomina enrutamiento.

EJEMPLO A: *Identificar la siguiente dirección IP en binario:*

11001100.00001000.00000000.10101010 (204.8.0.170)

La dirección de la máscara (MASK) es en binario :

11111111.11111111.11100000.00000000 (255.255.224.0)

Ahora la dirección se SubRed se toma la IP y considerando que todo lo que tenga 1s en la máscara se queda como esta en la IP, y todo lo que tenga 0s en la mascara se pone a 0 en la IP, la dirección de SubRed es:

11001100.00001000.00000000.00000000 (204.8.0.0)

EJEMPLO B: *La dirección IP en binario:*

00001001.01000011.00100110.00000000 (9.67.38.0)

Su máscara de red es:

11111111.11111111.11111111.11000000 (255.255.255.192)

Siguiendo el ejemplo anterior, la dirección de SubNet es:

00001001.01000011.00100110.00000000 (9.67.38.0)

En la dirección de la máscara de red, los último 6 bits han quedado a 0. Estos bits son los que definen las máquinas de la SubRed (2^6=64). De estas 64 máquinas quitamos la última de ellas (será para el Broadcast). Por tanto tendremos:

9.67.38.0 Subnet Address
9.67.38.1 1ª Máquina de la SubRed
9.67.38.2 2ª Máquina de la SubRed
…….
9.67.38.62 Ultima máquina de la SuRed
9.67.38.63 Broadcast

EJEMPLO C: *La dirección IP 201.222.5.121, la dirección de máscara 255.255.255.248, entonces, haciendo los correspondientes cálculos en binario tenemos que:*

201.222.5.121	***Dirección IP***
255.255.255.248	***Máscara***
201.222.5.120	***Dirección de Subred***

En la dirección de máscara, el 248 es 0111000, por tanto los últimos 3 bits a 0 son destinados para las máquinas de red (2^3=8), por tanto habrá 6 máquinas:

201.222.5.120	***Dirección de Subred***
201.222.5.121	***1ª Máquina de Subred***
201.222.5.122	***2ª Máquina de Subred***
……..	
201.222.5.126	***Ultima Máquina de Subred***
201.222.5.127	***Braodcast***

2.3 Interfaces

Este protocolo es utilizado por protocolos host-a-host, utiliza a su vez protocolos de red locales para llevar el datagrama internet a la próximo host de destino.

Un módulo TCP llamaría al módulo internet para tomar un segmento TCP (incluyendo la cabecera TCP y los datos de usuario) como la parte de datos de un datagrama internet. El módulo TCP suministraría las direcciones y otros parámetros de la cabecera internet al módulo internet como argumentos de la llamada. El módulo internet crearía entonces un datagrama internet y utilizaría la interfaz de la red local para transmitir el datagrama internet. En el caso de ARPANET, el módulo internet llamaría a un módulo de red local el cual añadiría el encabezado al datagrama internet creando así un mensaje ARPANET a transmitir al IMP. La dirección ARPANET sería deducida de la dirección internet por la interfaz de la red local y sería la dirección de algún host en ARPANET, el cual podría ser una pasarela a otras redes.

La descripción funcional de interfaces de usuario para IP es fcticia ya que cada sistema operativo dispondrá de distintos recursos, en consecuencia, diferentes implementaciones IP pueden tener diferentes interfaces de usuario. Sin embargo, todas deben proporcionar un cierto conjunto mínimo de servicios para garantizar que todas las implementaciones IP pueden soportar la misma jerarquía de protocolos.

El protocolo internet interactúa con la red local por un lado y con un protocolo de nivel superior o una aplicación por el otro. En lo que sigue, el protocolo de nivel superior o aplicación será llamado el "usuario", dado que es lo que usa el módulo

internet. Como el protocolo internet es un protocolo de datagramas, se mantiene un mínimo de memoria o estado entre transmisiones de datagramas, y cada llamada del usuario al módulo del protocolo internet proporciona toda la información necesaria para que el IP ejecute el servicio pedido.

Cuando el usuario envía un datagrama, ejecuta la llamada SEND (envío) suministrando todos los argumentos, el módulo del protocolo internet, al recibir esta llamada, comprueba los argumentos, prepara y envía el mensaje. Si los argumentos son válidos y el datagrama es aceptado por la red local, la llamada retorna con éxito. Si los argumentos no son correctos, o bien el datagrama no es aceptado por la red local, la llamada retorna sin éxito. En retornos de llamada sin éxito, se debe construir un informe razonable sobre la causa del problema, pero los detalles de tales informes corresponden a las distintas implementaciones.

Cuando un datagrama llega al módulo del protocolo internet desde la red local, o hay una llamada RECV(entrega) pendiente del usuario al que va dirigido o no la hay. En el primer caso, la llamada pendiente es satisfecha pasando la información desde el datagrama al usuario. En el segundo caso, se notifica al usuario de destino que tiene un datagrama pendiente. Si el usuario de destino no existe, se devuelve un mensaje de error ICMP al remitente, y los datos son desechados.

La notificación de un usuario puede ser a través de una pseudo interrupción o un mecanismo similar, correspondiente al entorno particular del sistema operativo de la implementación.

Una llamada RECV(entrega) de usuario puede ser inmediatamente satisfecha por un datagrama pendiente, o bien quedar pendiente hasta que llegue un datagrama. La dirección de origen se incluye en la llamada SEND(envío) en el caso de que el host remitente tenga varias direcciones (múltiples conexiones físicas o direcciones lógicas). El módulo internet debe comprobar que la dirección de origen es una de las direcciones legales de este host.

Una implementación también puede permitir o exigir una llamada al módulo internet para indicar interés en o reservar para uso exclusivo una clase de datagramas (p. ej., todos aquellos con un cierto valor en el campo protocolo), es funcionalmente una interfaz USUARIO/IP.

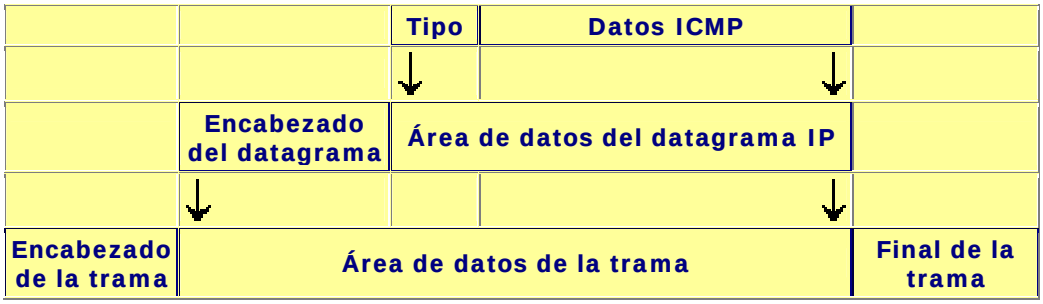
2.4 Relación con otros protocolos

En la capa de Intered, el protocolo IP se relaciona con otros protocolos de control que ayudan a fortalecer sus debilidades.

Protocolo ICMP (Internet Control Message Protocol, protocolo de mensajes de control y error), *proporciona un medio para transferir mensajes desde los dispositivos de encaminamiento y otros ordenadores a otro ordenador, debido a que el protocolo IP no es fiable, los datagramas pueden perderse o llegar defectuosos a su*

destino. ICMP se encarga de informar al origen si se ha producido algún error durante la entrega de su mensaje y de notificar posibles errores, al transportar distintos mensajes de control, pero esto no quiere decir que tome alguna decisión, esta tarea la realizan las capas superiores.

Cuando se construye un mensaje ICMP, se pasa a IP, que encapsula el mensaje con una cabecera IP, luego transmite el datagrama resultante.



Al ser el protocolo IP poco fiable puede darse el caso de que un mensaje ICMP se pierda o se dañe, en este caso no se creará un nuevo mensaje ICMP, sino que el primero se descartará.

La cabecera de un mensaje ICMP siempre contendrá los siguientes campos:

Tipo (8 bist): *especifica el tipo de mensaje ICMP.*

Código (8 bits): *especifica parámetros de mensaje que se pueden codificar en uno o unos pocos bits.*

Suma de Comproación (16 bits): *Se usa el mismo algoritmo de suma de comprobación que en IP.*

Parámetros (16 de bits): *se usa para especificar parámetros más largos.*

Los mensajes ICMP pueden ser del siguiente tipo:

CAMPO TIPO	TIPO DE MENSAJE ICMP
0	Respuesta de eco (Echo Reply)
3	Destino inaccesible (Destination Unreachable)
4	Disminución del tráfico desde el origen (Source Quench)
5	Redireccionar (cambio de ruta) (Redirect)
8	Solicitud de eco (Echo)
11	Tiempo excedido para un datagrama (Time Exceeded)
12	Problema de Parámetros (Parameter Problem)
13	Solicitud de marca de tiempo (Timestamp)
14	Respuesta de marca de tiempo (Timestamp Reply)
15	Solicitud de información (obsoleto) (Information Request)
16	Respuesta de información (obsoleto) (Information Reply)
17	Solicitud de máscara (Addressmask)
18	Respuesta de máscara (Addressmask Reply)

Echo Reply, Echo Request*, proporcionan información para comprobar que la comunicación entre dos entidades es posible, el receptor del mensaje echo esta obligado a devolver el mensaje de respuesta a echo, al mensaje echo se le asocia con un identificador y un número de secuencia. El comando* ***PING*** *envía mensajes de solicitud de eco a un host remoto e informa de las respuestas, en caso de existir comunicación:.*

1. *A envía un mensaje ICMP de tipo 8 (Echo) a B*
2. *B recibe el mensaje y devuelve un mensaje ICMP de tipo 0 (Echo Reply) a A*
3. *A recibe el mensaje ICMP de B y muestra el resultado en pantalla*

C:\ping 10.1.9.1 -n 1
Haciendo ping a 10.1.9.1 con 32 bytes de datos:
Respuesta desde 10.1.9.1: bytes=32 tiempo<10ms TDV=128

En la orden anterior el parámetro "-n 1" es para que el host A únicamente envíe 1 mensaje de solicitud de eco. Si no se especifica este parámetro se enviarían 4 mensajes (y se recibirían 4 respuestas). Si el host de destino no existiese o no estuviera correctamente configurado recibiríamos un mensaje ICMP de tipo 11 (Time Exceeded).

C:\ping 192.168.0.6 -n 1
Haciendo ping a 192.168.0.6 con 32 bytes de datos:
Tiempo de espera agotado.

Tiempo Excedido, *este mensaje se presenta cuando el tiempo de vida de un datagrama ha expirado, el ordenador enviará el mensaje si no se pudo completar la comunicación en el tiempo determinado. El comando* ***TRACERT (traceroute)*** *hace una traza a un determinado host. TRACERT funciona enviando mensajes ICMP de solicitud de eco con distintos TTL; traceroute, en cambio, envía mensajes UDP. Si la comunicación extremo a extremo no es posible, la traza nos indicará en qué punto se ha producido la incidencia.*

C:\tracert master

Tracing route to master [10.1.1.1]
over a maximum of 30 hops:

1 523 ms 340 ms 354 ms 10.1.1.71
2 334 ms 341 ms 340 ms MASTER [10.1.1.1]

Trace complete.

TimeStamp Request, TimeStamp Reply, *son similares a eco request y Echo Reply, pero además se registra en el paquete el instante en el que se emite, cuando el destinatario lo recibe y cuando lo devuelve*

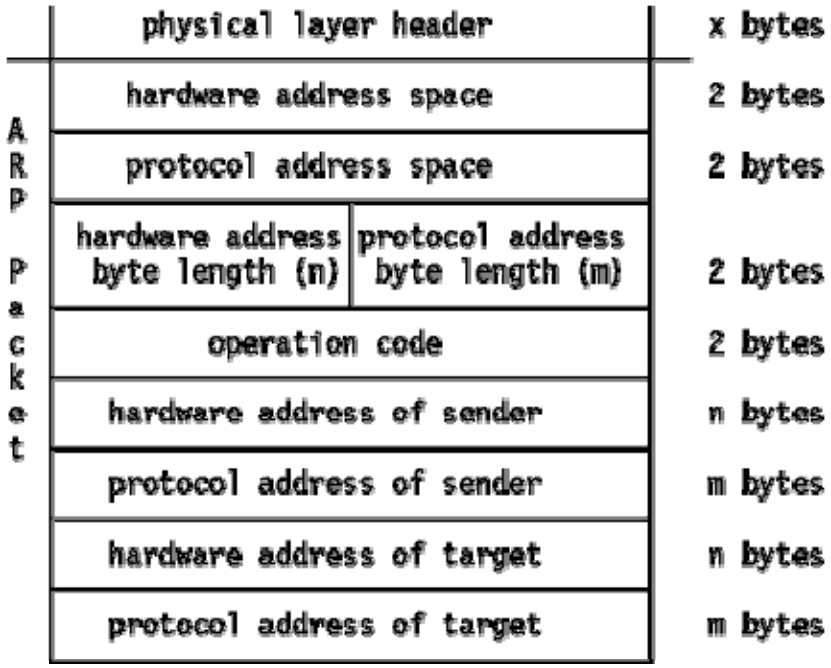
Redirect, *alerta al host emisor de que posiblemente un paquete se está encaminando incorrectamente.*

Parameter Problem, *envía un mensaje al emisor cuando el router detecta un valor ilegal en la cabecera.*

Protocolo ARP (Address Resolution Protocols), *Se encarga de hacer llegar los datagramas a la red destino, de acuerdo a rutas perfectamente establecidas, pero dentro de una red broadcast es necesario un mecanismo que permita identificar a que dirección MAC corresponde la dirección IP del paquete que se quiere entregar. Esto no se puede hacer por medio de una tabla estática, por que puede variar por un simple cambio de tarjeta o de ordenador.*

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.



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 físicas del 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.

EJEMPLO: *Visualiza entradas actuales del ARP interrogando a la corriente datos del protocolo. Si se especifica el inet_addr. el IP y la comprobación los direccionamientos para solamente el ordenador especificado se visualizan. En este caso se ven todas las entradas.*

C:\arp –a

Interface: 10.1.1.155 --- 0x10003

Internet Address	*Physical Address*	*Type*
10.1.1.1	*00-02-55-97-37-53*	*dynamic*
10.1.1.5	*00-60-94-23-5c-a6*	*dynamic*
10.1.1.6	*00-60-b0-6b-0e-37*	*dynamic*
10.1.1.19	*08-00-3e-00-f7-17*	*dynamic*
10.1.1.20	*00-01-4e-00-22-eb*	*dynamic*

***Protocolo IGMP (Internet Group Management Protocol)**, Es la norma para la multiencapsulación IP en Internet y se utiliza para establecer la membresía de la computadora anfitrión en grupos específicos de multiencapsulación en una sóla red. Los pormenores del protocolo le permiten a la computadora anfitrión informar a su enrutador local que utiliza los informes de membresía de la computadora anfitrión, acerca de su deseo de recibir los mensajes dirigidos a un grupo específico de multiencapsulación*

ICMP, hace uso de la difusión propia de una LAN para proporcionar un intercambio de información entre los ordenadores y los dispositivos de encaminamiento.

Los mensajes ICMP se transmiten en datagramas IP con los siguientes campos:

***Versión,** versión del protocolo.*

***Tipo,** el tipo 1, se refiere a una petición de un dispositivo de encaminamiento de multifusión, y el tipo 2, información enviada por un ordenador.*

***Suma de Comprobación,** es un código de detección de errores, dado por la suma complemento a 1 de todas las cuatro palabras de 16 bits del mensaje y se inicializa en 0.*

***Dirección de grupo,** el valor cero en un mensaje de solicitud y una dirección de grupo válida en un mensaje de informe.*

El objetivo de que un ordenador utilice ICMP es hacerse conocer como miembro de un grupo con una dirección multifusión dada a otros ordenadores en la red y a los dispositivos de encaminamiento. Para ser parte del grupo, el ordenador envía un mensaje con el campo de dirección de grupo como la dirección de multidifusión, el mensaje se envía a un datagrama IP con la misma dirección; todos los ordenadores que son parte del grupo reciben el mensaje y acogen al nuevo miembro, al igual que los dispositivos de encaminamiento conectados deben atender las direcciones IP de multidifusión para recibir los mensajes.

Un dispositivo de encaminamiento esta constantemente enviando mensajes de petición de direcciones y los ordenadores deben responder.

Protocolo RSVP(Resource ReSerVation Protocol), *RSVP es un protocolo que se desarrolla entre los usuarios y la red, y entre los diferentes nodos (routers) de la red que soportan este protocolo. Consiste en hacer «reservas» de recursos en dichos nodos para cada flujo de información de usuario, con la consecuente ocupación de los mismos. Esto requiere, intercambio de mensajes RSVP entre dichos entes funcionales, así como «mantener» estados de reserva en cada nodo RSVP. De manera que tanto la solicitud de las reservas, como el mantenimiento de éstas durante la comunicación, y la posterior cancelación, implica el intercambio de mensajes de señalización, lo que representa un tráfico considerable cuando de entornos como Internet se trata.*

RSVP es un protocolo señalización de QoS, y posibilita: dar a las aplicaciones una modo uniforme para solicitar determinado nivel de QoS, además, encontrar una forma de garantizar cierto nivel de QoS, y proveer autentificación.

RSVP ofrece dos tipos de servicios, a saber: Servicio de carga controlada y servicio garantizado, en el primero la pérdida de paquetes debe ser muy baja o nula, y en el segundo solicita cierto ancho de banda y cierta demora de tránsito, siendo este último el más complejo de implementar.

RSVP utiliza dos tipos de mensajes básicos: RESV y PATH, a través de los cuales se lleva a cabo la reserva de recursos en la red previo a la comunicación. El mensaje PATH se utiliza para proporcionar información de encaminamiento hacia arriba, en los protocolos de encaminamiento multidifusión solo se mantiene la ruta hacia abajo, sin embargo, los mensajes RESV se deben propagar hacia arriba a través de todos los dispositivos de encaminamiento intermedios y hacia todos los ordenadores de origen. Al presentarse la falta de información de encaminamiento inversa en los protocolos de encaminamiento, RESV proporciona esto con los mensajes Path, el ordenador que participa como origen de grupo multidifusión, emite un mensaje Path que se transmite a través del árbol de distribución a todos los destinos y a lo largo de todo el camino, cada dispositivo de encaminamiento y ordenador destino crea un estado de camino que indica el salto inverso que hay utilizar para ese origen.

Otros mensajes del protocolo RSVP son:

- *PATHTEAR: son mensajes generados por la fuente de datos de usuario para eliminar los estados path´s en todos los routers RSVP. Siguen la misma ruta que los mensajes PATH´s. También pueden ser originados por cualquier nodo cuando se agota el timeout del estado path.*
- *RESVTEAR: son generados por los receptores para borrar los estados de reserva en los routers RSVP, por tanto viajan en el sentido upstream. Pueden ser también originados por nodos RSVP al agotarse el timeout del estado de reserva de los mismos.*
- *PATHERR: viajan en sentido upstream hacia el emisor siguiendo la misma ruta que los mensajes PATH´s, y notifican errores en el procesamiento de mensajes PATH´s, pero no modifican el estado del nodo por donde ellos pasan en su «viaje» hacia la aplicación emisora.*

- *RESVERR: notifican errores en el procesamiento de mensajes RESV, o notifican la interrupción de una reserva. Se transfieren en la dirección downstream hacia el receptor o receptores apropiados.*

2.5 Operación y Descripción de Funciones

En este punto es importante recalcar de qué forma opera el protocolo IP para transmitir un datagrama de una aplicación a otra, asumiendo una pasarela intermedia:

- *La aplicación prepara sus datos y llama a su módulo internet local para enviar los datos como un datagrama y pasa la dirección de destino y otros parámetros como argumentos de la llamada.*

- *El módulo internet prepara una cabecera de datagrama y adjunta los datos, determina una dirección de la red de área local para esta dirección internet (dirección de una pasarela) y envía el datagrama y la dirección a la interfaz de red local, la interfaz crea una cabecera de red local, le adjunta el datagrama y entonces envía el resultado a través de la red.*

- *El datagrama llega a un host pasarela encapsulado en la cabecera de red local, la interfaz de red local desprende la cabecera y dirige el datagrama hacia el módulo internet, éste determina a partir de la dirección internet que el datagrama debe ser reenviado a otro host en una segunda red y determina una dirección de red local para el host de destino, luego llama a la interfaz de red local de esa red para enviar el datagrama.*

- *Esta interfaz de red local crea una cabecera de red local y le adjunta el datagrama enviando el resultado al host de destino, el cual le quita al datagrama la cabecera de red local y se lo pasa al módulo internet, determinando que el datagrama va dirigido a una aplicación en este host. Pasa los datos a la aplicación en respuesta a una llamada al sistema, pasando la dirección de origen y otros parámetros como resultado de la llamada.*

Una de las principales funciones del Protocolo IP es enrutar paquetes IP a través de un conjunto de redes TCP-IP interconectadas. Los paquetes IP son encaminados desde un computador a otro a través de redes individuales basándose en la interpretación de una dirección IP. Para cumplir con esta función el protocolo IP utiliza las direcciones IP fuente y destino que se encuentran en una cabecera IP a fin de determinar la ruta de los paquetes IP desde un computador fuente a un computador destino. Este proceso se conoce con el nombre de ***enrutamiento*** *de paquetes IP. Durante el proceso de enrutamiento, un paquete IP puede ser fragmentado cuando el tamaño del mismo es mayor que la máxima unidad de transferencia de datos - MTU Maximun Transfer Unit*

Los parámetros de control de una cabecera IP: Dirección IP fuente, Dirección IP destino, Identificación, Flags y Posición son uilizados para cumplir con las funciones de ***enrutamiento, fragmentación y reensamblaje*** *del protocolo IPv4.*

Al hablar de ***enrutamiento,*** *Se inicia con una distinción entre nombres, direcciones y rutas. El protocolo IP maneja direcciones y es tarea de los protocolos de mayor nivel (es decir, protocolos host-a-host o entre aplicaciones) hacer corresponder nombres con direcciones. El módulo internet hace corresponder direcciones de internet con direcciones de red local. La tarea de los procedimientos de menor nivel (es decir, redes locales o pasarelas) es realizar la correspondencia entre direcciones de red local y rutas.*

Las direcciones son de una longitud fija de 32 bits, comienza por un número de red, seguido de la dirección local, siempre tomando en cuenta las principales clases de direcciones internet.

Se debe tener cuidado al relacionar direcciones internet con direcciones de red local; un host individual físicamente hablando debe ser capaz de actuar como si fuera varios hosts distintos, hasta el punto de usar varias direcciones internet distintas. Algunos hosts tendrán también varios interfaces físicos (multi-homing), esto quiere decir que se debe establecer algún mecanismo que permita a un host tener varios interfaces físicos de red, cada uno de ellos con varias direcciones lógicas internet.

IP soporta operaciones de ***fragmentación,*** *por la que una unidad de datos de protocolo (PDU) se divide y segmenta en unidades más pequeñas. Es una característica que puede ser muy útil, ya que no todas las redes utilizan PDU del mismo tamaño. Por ejemplo, las redes de cobertura amplia (WAN) basadas en X.25 utilizan típicamente PDU (denominadas paquetes en el contexto X.25) con un campo de datos de 128 octetos, y Ethernet limita el tamaño de una PDU a 1500 octetos*

Sin el uso de fragmentación, las pasarelas emplearían recursos para intentar resolver incompatibilidades de los tamaños de las PDU de las diferentes redes. IP resuelve el problema estableciendo unas reglas de fragmentación en la pasarela, y de reconstrucción en el computador receptor.

Un datagrama internet puede ser marcado como "no fragmentar", en este caso el datagrama no será fragmentado entre distintas redes bajo ninguna circunstancia y si éste datagrama no puede ser entregado en su destino sin fragmentarlo se descarta.

El procedimiento de ***fragmentación y reensamblaje*** *en internet tiene que ser capaz de dividir un datagrama en un número casi arbitrario de piezas que puedan ser luego reensambladas. El receptor de los fragmentos utiliza el campo de identificación para asegurarse de que no se mezclan fragmentos de distintos datagramas. El campo posición ("offset") le indica al receptor la posición de un fragmento en el datagrama original. La posición y longitud del fragmento determinan la porción de datagrama original comprendida en este fragmento. El indicador "más-fragmentos" indica el*

último fragmento. Estos campos proporcionan información suficiente para reensamblar datagramas.

El campo identificador se usa para distinguir los fragmentos de un datagrama de otro. El módulo de protocolo de origen de un datagrama internet establece el campo identificador a un valor que debe ser único para ese protocolo y par origen-destino durante el tiempo que el datagrama estará activo en el sistema internet. El módulo de protocolo de origen de un datagrama completo pone el indicador "más-fragmentos" a cero y la posición del fragmento a cero.

Para fragmentar un datagrama internet grande, un módulo de protocolo internet crea dos nuevos datagramas internet y copia el contenido de los campos de cabecera internet del datagrama grande en las dos cabeceras nuevas, los datos son divididos en dos tomando una resolución mínima de 8 octetos (64 bits) (la segunda parte puede no ser un múltiplo entero de 8 octetos, pero la primera sí debe serlo). El número de bloques de 8 octetos en la primera parte NFB (Number of Fragment Blocks: Número de Bloques del Fragmento), es colocado en el primer nuevo datagrama y el campo longitud total se establece a la longitud del primer datagrama, el indicador "más fragmentos" es puesto a uno. La seguda parte de los datos es colocada en el segundo nuevo datagrama y el campo longitud total se establece a la longitud del segundo datagrama. El indicador "más fragmentos" lleva el mismo valor que en el datagrama grande. El campo posición del segundo nuevo datagrama se establece al valor de ese campo en el datagrama grande más NFB. Todo este procedimiento puede generalizarse para una n-partición, mucho mejor que para la división en dos partes.

Para ensamblar los fragmentos de un datagrama, un módulo de protocolo internet (por ejemplo en un host de destino) combina todos los datagramas internet que tengan el mismo valor en los cuatro campos: identificación, origen, destino y protocolo. La combinación se realiza colocando la parte de los datos de cada fragmento en su posición relativa indicada por la posición del fragmento en la cabecera internet de ese fragmento. El primer fragmento tendrá posición cero, y el último fragmento tendrá el indicador "más fragmentos" puesto a cero.

2.6 Protocolo IPv6

Al parecer el tiempo de vida útil del Protocolo IP4 esta llegando a su fin, por su limitación en el campo de 32bits, que en un principio al disponer de 2^{32} direcciones diferentes, cerca de 4.000 millones de direcciones posibles, parecería suficiente para las necesidades de internet, pero ahora hay algunas razones que han llevado a los investigares a mejorar esta versión:

- *El modelo de direccionamiento IP, requiere un número de red único aunque no este conectado a internet.*
- *Alto crecimiento en redes, redes inalámbricas, y el mismo internet avanza a pasos agigantados.*
- *La forma poco económica de asignar direcciones, por su estructura (nùnero de red y de ordenador), una vez que se le asigna la dirección a la red, pueden o no ser usados todas las direcciones.*

- *El creciente uso de TCPIP en el mercado, como para interconectar terminales electrónicos de puntos de venta, receptores de TV por cable, etc.*

Por estas y más razones, actualmente está en estudio IP v6 (también llamada IPng=Internet Protocol Next Generation), que introduce numerosos cambios respecto a la versión actual.

*La diferencia principal es que se pasa de direcciones IP de 32 bits a direcciones de 128 bits (16 bytes), con lo que se eliminan las restricciones del sistema actual, al disponerse de 7*1023 direcciones IP por metro cuadrado en el planeta Tierra. También se ha mejorado la seguridad del protocolo y la cabecera de los datagramas (que ahora está mejor estructurada y que consta de 320 bits - múltiplos de 8 bytes).*

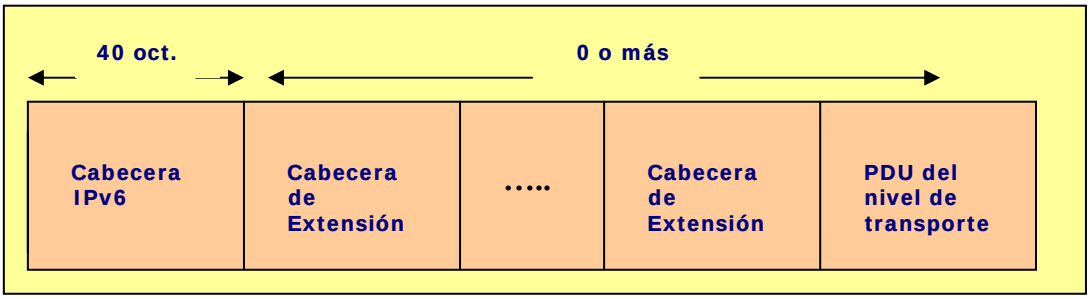
Las nuevas direcciones IP identifican a un interfaz o conjunto de interfaces, y no a un nodo, como lo hace la versión actual de IP, y el número de direcciones que permite es mucho más elevado, del orden de 2^{128}, lo que da más de 665.000 trillones de direcciones posibles, aunque este número se verá disminuido cuando se les aplique a las direcciones una estructuración jerárquica adecuada. Con IPv6 se persiguen los siguientes fines:

- *Espacio de direcciones ampliado*
- *Aumento en la flexibilidad de direccionamiento*
- *Mecanismo de Opciones mejorado*
- *Soportar miles de millones de host.*
- *Reducir el tamaño de las tablas de ruteo.*
- *Simplificar el protocolo IP, lo que permitirá un procesamiento más rápido.*
- *Proveer más seguridad al sistema.*
- *Diferentes tipos de servicio.*
- *Mejorar el multicasting.*
- *Permitir que el protocolo pueda cambiar en el futuro, permitiendo a la vez la compatibilidad de los protocolos nuevos con los antiguos.*

IP v6 no usará la fragmentación de paquetes en los router´s, estando estos limitados al manejo de paquetes de 576 bytes como máximo. Si un paquete es mayor que este límite será rechazado por el router, quedando bajo la responsabilidad del host el fragmentarlo de forma adecuada para su transmisión.

También se elimina el checksum (suma de comprobación de errores), y se permite el uso de datagramas de tamaños excepcionalmente grandes, conocidos como jumbograms, para su uso en aplicaciones de supercomputadores.

En la estructura de IPv6, la cabecera tiene una longitud fija de 40 octetos y se han definido además cabeceras de extensión:



Cabecera de Opciones de Salto, define opciones especiales que requieren procesamiento en cada salto.

Cabecera de encaminamiento, proporciona un encaminamiento ampliado.

Cabecera de fragmentación, contiene información de fragmentación y reensamblaje.

Cabecera de autentificación, proporciona la integridad del paquete y su autentificación.

Cabecera de encapsulamiento de la carga de seguridad, proporciona seguridad.

Cabecera de las opciones para el destino, contiene información opcional para ser examinado en el nodo destino.

La cabecera consta de los siguientes campos:

Versión (4 bits), versión del protocolo.

Clase de Tráfico (8 bits), disponible para el uso del nodo origen y los dispositivos de encaminamiento de reenvío para identificar y distinguir entre diferentes clases o prioridades.

Etiqueta de Flujo (20 bits), etiqueta los paquetes que requieren un tratamiento especial en los dispositivos de encaminamiento dentro de la red.

Longitud de la Carga útil (16 bits), representa la longitud total de todas las cabeceras más la PDU de la capa de transporte.

Cabecera siguiente (8 bits), identifica el tipo de cabecera que sigue inmediatamente a la cabecera IPv6.

Límite de saltos (8 bits), el número de saltos permitidos para este paquete.

Dirección origen (128 bits), dirección del productor del paquete.

Dirección destino (128 bits), dirección del destino del paquete.

Como primera clasificación de las direcciones propuesta por IP v6 cabe destacar la división basada en si se identifica una interfaz o un conjunto de ellas, lo que da lugar a los siguientes tipos:

- ***Direcciones unicast**: son aquellas que identifican un único interfaz en la red.*
- ***Direcciones anycast**: que identifican a un grupo de interfaces de la red, de tal forma que el paquete se enviará a una cualquiera de las intefaces que componen el grupo. Su formato es análogo al de las direcciones unicast, ya que estas direcciones son en realidad direcciones unicast asignadas a varios interfaces.*
- ***Direcciones multicast**: que identifican a un conjunto de interfaces, de manera que el paquete es enviado a cada una de ellas individualmente. Esto permite eliminar las direcciones de broadcast, ya que realizan la misma función.*

3. Aplicaciones IP

3.1 Servicio de Nombres: DNS

DNS (Domain Name Service) está diseñado como un sistema de base de datos distribuido que permite convertir nombres de dominio en sus direcciones IP correspondientes. Cada organización ejecuta sus propios servidores DNS y mantiene los registros de la base de datos de asignación de nombres, o registros de recursos, para su dominio, por ello al realizar una solicitud de resolución de nombres, un servidor DNS comprueba primero si en sus registros existe la dirección IP correspondiente y si no tiene la respuesta, pedirá la información a otros servidores DNS.

Los nombres simbólicos se agrupan en zona, donde uno o más hosts tienen la tarea de mantener la base de datos de nombres simbólicos y direcciones IP y de suministrar la función de servidor para los clientes que deseen hacer las traduciones. Estos servidores de nombres locales se interconectan lógicamente en un 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. 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 la ejerce el mismo servidor.

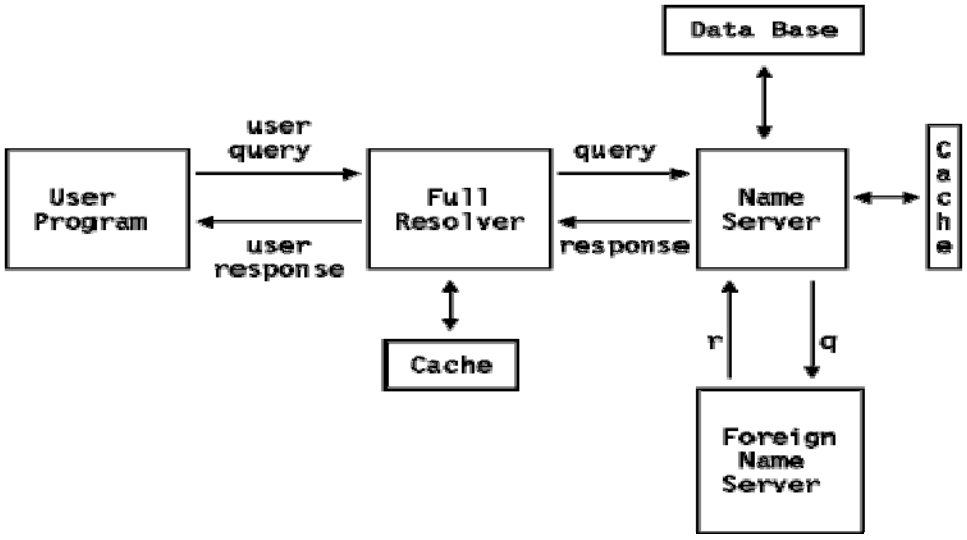
Se pueden por ejemplo, seguir los pasos que realiza un navegador para ingresar al sitio www.yahoo.com al intentar resolver la dirección IP del nombre de dominio:

- *El navegador Web llama al cliente DNS e intenta resolver el nombre localmente utilizando información almacenada en caché de una consulta anterior.*
- *Si no se puede resolver localmente, el cliente pide una respuesta a un servidor DNS conocido. Si ese servidor DNS nunca ha atendido solicitudes del mismo nombre de dominio, "www.yahoo.com" (dentro de un cierto periodo de tiempo), recupera la dirección IP correspondiente de su caché y la devuelve al cliente.*
- *Si ese servidor DNS no encuentra la respuesta, el cliente puede preguntar a uno de los servidores DNS raíz globales, que mantienen y devuelven punteros de los servidores DNS autorizados para los dominios de nivel superior. En este caso, se devuelve al cliente la dirección IP del servidor autorizado para el dominio "com".*
- *Del mismo modo, el cliente pregunta al servidor de "com" dónde está el servidor "yahoo.com". El cliente pasa entonces la consulta original al servidor "yahoo.com".*
- *Como el servidor "microsoft.com" mantiene localmente los registros autorizados del dominio, devuelve el resultado final al cliente y completa la búsqueda de la dirección IP específica.*

3.1.1 Resolución de Dominios: Nombres y Direcciones

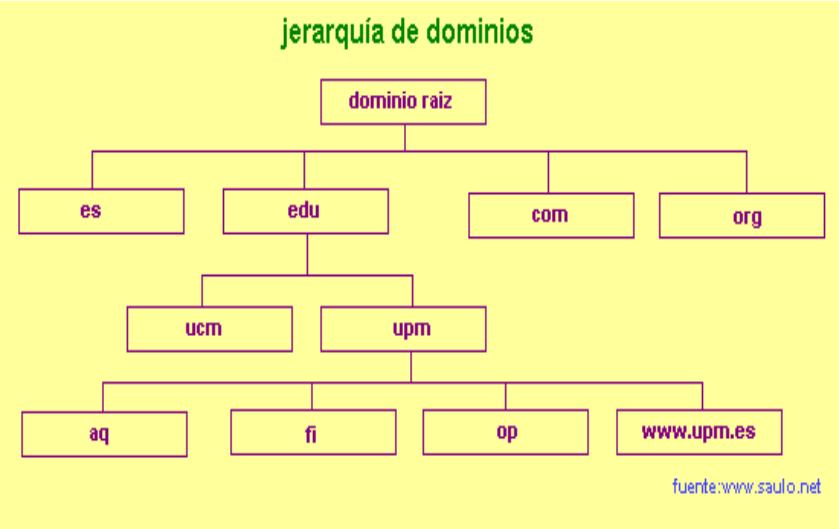
La resolución de nombres de dominio es la traducción de un nombre de dominio a su correspondiente dirección IP, que no es otra cosa que un proceso cliente/servidor. Cuando un cliente DNS desea obtener la IP asociada a un nombre de dominio concreto puede preguntar servidor DNS de dos formas diferentes: en forma ***recursiva,*** *cuando servidor DNS debe intentar por todos los medios posibles obtener la resolución pedida, aunque para ello tenga que preguntar a otros servidores DNS. Este tipo de preguntas es el que suelen hacer los host al servidor DNS local de su proveedor de Internet a otros servidores DNS de rango supeior cuando no encuentra en su memoria caché la dirección IP asociada al nombre de dominio por el que le preguntamos.; o en forma* ***interactiva,*** *en las que el servidor devolverá al cliente la resolución pedida en caso de conocerla, y en caso contrario le devolverá la dirección IP de otro servidor DNS que sea capaz de resolver el nombre solicitado.*

El gráfico presenta un programa denominado "full resolver", distinto del programa de usuario, que envía todas las peticiones al servidor de nombres. El servidor de nombres guarda en la memoria cache las respuestas para su uso en el futuro.



Uso del "full resolver" para la resolución de nombres de dominio

Un dominio es sólo un índice dentro de la base de datos de DNS, puede ser una máquina o puede ser un nodo del cual pueden partir otros dominios (o las dos cosas a la vez).



Los nombres son esencialmente rutas en un árbol, como se aprecia en el gráfico. El árbol de DNS puede ramificarse de diversas maneras en cada punto de intersección, llamado nodo. Cada nodo puede tener hasta 63 caracteres de longitud. El dominio raíz tiene una etiqueta (de tamaño cero), la cual es reservada. El nombre completo de un nombre de dominio es una secuencia de etiquetas en la ruta desde ese nodo hasta la raíz. Cuando el dominio raíz aparece por si mismo es denominado "." por conveniención. De manera que cuando alguien escribe una dirección terminada con un punto ésta es interpretada como una dirección absoluta.

Desde el dominio raíz parten los dominios del primer nivel, luego del segundo nivel, y así sucesivamente. Cada uno de los dominios puede contener tanto host particulares como más subdominios. En el gráfico se parte del dominio "edu", correspondiente a organizaciones y entidades educativas, se deriva el dominio "upm", correspondiente a la Universidad Politécnica de Madrid, y de éste parten diferentes subdominios, como "aq" que corresponde a la Escuela de Arquitectura, y cada uno de los dominios es gestionado por un servidor independiente.

Cuando la capa IP de un host concreto necesita saber la dirección IP de una serie de paquetes a partir de los nombres de dominio se establece una conexión UDP (User Datagram Protocol) con el servidor DNS adecuado, que le da la equivalencia necesaria.

Actualmente cada servidor DNS gestiona y actualiza los nombres de host de un dominio o subconjunto de nodos de Internet que son administrados por un organismo. De esta forma, cuando se conecta un nuevo nodo a Internet, su nombre de host es dado de alta en el servidor DNS del dominio al que corresponda.

Los dominios de un nodo van separados por puntos y organizados de forma jerárquica, empezando por el dominio de mayor nivel, por ejemplo:

http://www.uazuay.edu.ec/maestrias/sistfmla.htm

protocolo web dominio directorio del servidor fichero

Los dominios están clasificados en función de su estructura organizativa como geográficamente, los más importantes en función de su estructura ***organizativa*** *son:*

- *.com - organización comercial*
- *.edu - instituciones educativas y universidades*
- *.gov - organizaciones gubernamentales*
- *.mil - organizaciones militares*
- *.net – organizaciones nacionales de redes*
- *.org - organizaciones no comerciales*

Aunque actualmente se ha propuesto la creación de nuevos dominios como:

- *store - negocios que ofrecen bienes para comprar*
- *web - entidades que enfatizan actividades en el web*
- *art - entidades con actividades recreacionales y de entretenimiento*
- *info - entidades que proveen servicios de información*
- *name - aquellos que desean una nomenclatura personal o individual*
- *aero - empresas de aviación*
- *coop - cooperativas*
- *museum - museos*

Mientras que en función de su localización geográfica tenemos: ec (Ecuador), arg. (Argentina), es (España), .fr (francia), .it (Italia), etc.

3.1.2 Elementos del DNS

Para su funcionamiento, el sistema DNS utiliza tres componentes principales:

- ***Espacio de nombres de dominio (domain name space)****: base de datos distribuida entre distintos servidores.*

- ***Servidores DNS (name servers)****: servidores especiales que contestan a las peticiones de los clientes, consultando para ello sus bases de datos de resolución. En caso de no disponer de la equivalencia solicitada por el cliente pueden reenviar la petición a otro servidor DNS.*

- ***Clientes DNS (resolvers)****: que son host particulares (estaciones de trabajo o servidores) que envían peticiones de resolución de nombres a un servidor DNS.*

3.1.3 Funcionamiento del Resolver de Nombres de Dominio

Un resolver es un conjunto de bibliotecas de las aplicaciones clientes (Es decir aquellas que solicitan información acerca de un espacio de dominios de nombres). Sus funciones son:

- *Interrogar al servidor de nombres*
- *Interpretar respuestas*
- *Devolver información al programa que la solicita.*

Los servidores de nombres tienden a buscar datos de un espacio de dominio de nombre. Tienen que comportarse de esa manera, dada la poca inteligencia del resolver. No sólo pueden dar datos acerca de zonas de la que tienen autoridad sino que pueden buscar a lo largo de un espacio de dominio para encontrar datos sobre los que no tienen autoridad (A esto se le conoce como resolución).

La resolución comienza siempre desde los servidores de dominios superiores hasta llegar al servidor que tiene la información autorizada de un dominio en particular (Es decir comienza desde la parte superior del árbol invertido hasta llegar a la rama buscada).

Al ser las peticiones sobre nombres de dominio de dos tipos: recursivas o iterativas, para poder cumplir con las funciones, 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 haga 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.

Así mismo laas 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, guardará la información en su memoria cache para mejorar el*

- *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.*

3.1.4 Registro de Recursos de Dominio

La base de datos distribuida del DNS se compone de RRs("resource records"). Estos proporcionan una mapeo entre nombres de dominio y objetos de red. Los objetos de red más comunes son las direcciones de los host, 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

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.*

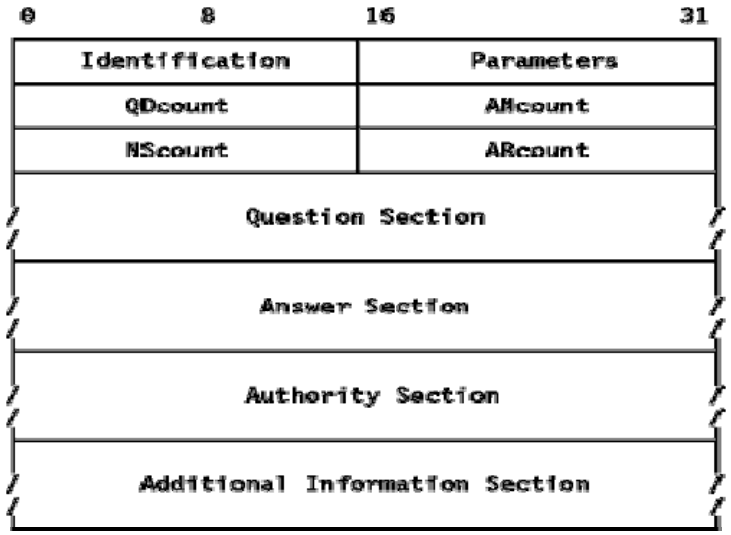
Class, *Identifica la familia del protocolo. Valores habituales son:* ***IN,*** *Sistema de Internet y* ***CH,*** *Sistema Chaos.*

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.*

Rdata, *El valor depende del tipo, por ejemplo:* ***A,*** *Una 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.*

3.1.5 Mensajes de DNS

Los mensajes del DNS tienen el siguiente formato:



En una consulta la cabecera y la parte "question" se usan cuando el "resolver" envía la trama al servidor de nombres y las respuestas o retransmisiones de las consultas usan la misma trama, pero llenan más secciones ("answer/authority/additional").

La sección de cabecera es fija, tiene una longitud de 12 bytes, el resto de secciones tienen 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 specificando 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".*

Formato del campo "Question" 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".*

En las Secciones "Answer", "Authority" y "Additional Resource" contienen un número variable de registros de recursos. El número se especifica en el campo correspondiente de la cabecera y los campos que preceden al TTL tienen el mismo significado que en la sección "question".

- ***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".*

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: estos 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. Si el valor es cero especifica el primer byte del campo ID de la cabecera.*
- *Si se usa compresión en un campo en la parte de "Rdata" de una sección "answer", "authority" o "additional", el campo "RDlength" precedente contiene, después de haber efectuado la compresión, la longitud real.*

3.1.6 Aplicaciones de DNS

Hay algunas utilidades que pueden ayudar en el manejo de DNS, como administradores de BIND. Vamos va ver algunas de ellas:

***hostcvt**, ayuda con la configuración inicial de BIND convirtiendo su fichero /etc/hosts en ficheros maestros para **named**. Genera tanto las entradas del mapeado directo (A) como el mapeado inverso (PTR), y toma en cuenta los alias. Por supuesto, no hará todo el trabajo por usted, así que todavía tendrá que ajustar los palores de temporización el el registro SOA, por ejemplo, a añadir registros MX. También, le puede ayudar a ahorrarse algunas aspirinas. **hostcvt** es parte de las fuentes BIND, pero puede encontrase como un paquete individual en unos pocos servidores FTP.*

*Tras la puesta en marcha del servidor, normalmente habrá que probarla. Algunas utilidades facilitan la vida para esto. Una de ellas es **dnswalk**, que está basada en Perl. La segunda es **nslint**. Ambas recorren la base de datos DNS buscando errores habituales y verificando que la información que encuentran es consistente. Otras dos utilidades interesantes son **host** y **dig**, que vienen con el paquete BIND y pueden usarse para una inspección manual del las bases de datos.*

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.*

3.1.7 Caso Práctico

La herranienta ***nslookup,*** *es original de BIND, puede utilizarse para resolver problemas con los servidores DNS.* ***Nslookup*** *se utiliza desde una ventana de interfaz de comandos. Dispone de dos modos: uno interactivo y otro no interactivo.*

Las ***Consultas no interactivas,*** *el nslookup las resuelve de nombre a dirección, como en este caso:*

C:\>nslookup mozart
Server: mozart.hoople.edu
Address: 200.250.199.4
Name: mozart.hoople.edu
Address: 200.250.199.4

Las ***Consultas interactivas,*** *el nslookup la resuelve de dirección a nombre, donde debe introducir in-addr.arpa y consulta los registros PRT y no los registros de dirección.*

C:\>nslookup 200.190.50.4
Server: mozart.hoople.edu
Address: 200.250.199.4
**** mozart.hoople.edu can't find 200.190.50.4: Non existent domain*

En caso de que el nslookup no consiga conectarse al servidor, realiza varios intentos antes de pasar al siguiente, y mientras tanto presenta mensajes de TimeOut.

C:\>nslookup mozart
DNS request timed out.
Timeout was 2 seconds.
DNS request timed out.
Timeout was 4 seconds.
DNS request timed out.
Timeout was 8 seconds.
**** Imposible encontrar el nombre de servidor para la dirección*

200.250.199.4: Timed out
**** Los servidores predeterminados no están disponibles*
Server: UnKnown
Address: 200.250.199.4
**** La petición a UnKnown ha expirado*

El siguiente ejemplo muestra el principio de una sesión interactiva. El carácter > es el símbolo de entrada de nslookup.

C:\>nslookup
Default Server: mozart.hoople.edu
Address: 200.250.199.4
>

Nslookup admite varias órdenes y consultas. Las más básicas son las consultas de nombre y dirección:

> mozart
Server: mozart.hoople.edu
Address: 200.250.199.4
Name: mozart.hoople.edu
Address: 200.250.199.4

El objetivo de una búsqueda queda definido por la variable querytype (q). Su valor predeterminado es a, que equivale a buscar en los registros de dirección. El siguiente listado muestra cómo examinar el registro SOA de un dominio, El primer paso consiste en utilizar la orden set q=soa para limitar el objetivo de la consulta a los registros SOA.

> set q=soa
> hoople.edu
Server: mozart.hoople.edu
Address: 200.250.199.4
hoople.edu
primary name server = mozart.hoople.edu
responsible mail addr = peters.hoople.edu
serial = 15
refresh = 3600 (1 hour)
retry = 600 (18 mins)
expire = 86400 (1 day)
default TTL = 3600 (1 hour)

El registro MX muestra dos consultas de mail1, al cambiar el valor de querytype a mx, el resultado presenta el contenido del registro MX.

> set q=mx
> mail1
Server: mozart.hoople.edu
Address: 200.250.199.4

mai11.hoople.edu MX preference = 10, mail exchanger = mai11.hople.edu
mai11.hoople.edu internet address = 200.190.50.254

La variable querytype, puede examinar todos los registros de un host estableciendo el valor any.

> set q=any
> mai11
Server: mozart.hoople.edu
Address:200.250.199.4
mail1.hoople.edu internet address = 200.190.50.254
mail1.hoople.edu MX preference = 10, mail exchanger = mai11.hoople.edu
mai11.hoople.edu internet address = 200.190.50.254

Para revisar los registros de un dominio, se puede acudir a la orden 1s que acepta varios argumentos que restringen o amplían la consulta y el parámetro d ayuda en este caso a presentarlos

> ls -d hoople.edu
hoople.edu. WINS
WINS lookup info
flags =0 ( )
lookup timeout = 5
cache TTL = 600
server count = 1
WINS server = (200.190.50.4)

hoople.edu.	*NS*	*mozart.hoople.edu*
ftp	*CNAME*	*jsbach,hoople.edu*
haydn	*A*	*200.190.50.1*
haydn	*A*	*200.250.199.1*
brubeck.jazz	*A*	*200.250.199.81*
ellington.jazz	*A*	*200.250.199.80*
parker.jazz	*A*	*200.250.199.82*
jsbach	*A*	*200.250.199.3*
mail1	*A*	*200.190.50.254*
mai11	*MX*	*10 mai11.hoople.edu*
mozart	*A*	*200.250.199.4*
papa	*CNAME*	*haydn.hoople.edu*
papa190	*A*	*200.190.50.1*
papa250	*A*	*200.250.199.1*
schubert	*A*	*200.190.50.4*
hoople.edu	*SOA*	*mozart.hoople.edu*
peters.hoople.edu.		
		(15 3600 600 86400 3600)

El parámetro -t. acepta un argumento, un nombre de registro de recurso o any, y configura las consultas de manera similar a la variable querytype El siguiente

ejemplo sólo muestra los registros de dirección, además encuentra los registros de dirección de los subdominios.

```
> ls -t a hoople.edu.
[mozart.hoople.edu]
hoople.edu.                NS      server = mozart.hoople.edu
haydn.hoople.edu.          A       200.190.50.1
haydn.hoople.edu.          A       200.250.199.1
brubeck.jazz.hoople.edu.   A       200.250.199.81
ellington.jazz.hoople.edu. A       200.250.199.80
parker.jazz.hoople.edu.    A       200.250.199.82
jsbach.hoople.edu.         A       200.250.199.3
mail1.hoople.edu.          A       200.190.50.254
mozart.hoople.edu.         A       200.250.199.4
papa190.hoople.edu.        A       200.190.50.1
papa250.hoople.edu.        A       200.250.199.1
schubert.hoople.edu.       A       200.190.50.4
```

Puede enviar todos los argumentos presentados con la orden ls a un archivo, y para revisar el archivo sin Salir del nslookup puede ejecutarse la orden view

```
> ls -t a hoople.edu > temp
[mozart.hoople.edu]
Received 18 records.

> view temp
brubeck.jazz.hoople.edu.   A       200.250.199.81
ellington.jazz.hoople.edu. A       200.250.199.80
haydn.hoople.edu.          A       200.190.50.1
haydn.hoople.edu.          A       200.250.199.1
hoople.edu.                NS      server = mozart.hoople.edu
jsbach.hoople.edu.         A       200.250.199.3
mail1.hoople.edu.          A       200.190.50.254
mozart.hoople.edu.         A       200.250.199.4
papa190.hoople.edu.        A       200.190.50.1
papa250.hoople.edu.        A       200.250.199.1
parker.jazz.hoople.edu.    A       200.250.199.82
schubert.hoople.edu.       A       200.190.50.4
[mozart.hoople.edu]
```

Existe la posibilidad de consultar un servidor local con la orden lserver, o un servidor de nombres predeterminado con la orden server.:

```
> server schubert.hoople.edu
Server: schubert.hoople.edu
Address: 200.190.50.4
```

3.2 Correo Electrónico: Protocolos

El correo electrónico es un servicio básico de Internet, que en un principio es igual que la mensajería postal, pero el electrónico permite una comunicación rápida, cómoda y barata, que además de mensajes puede intercambiar programas, audio, vídeo e imágenes. Un sistema de correo electrónico define un buzón que contiene los archivos donde son almacenados los mensajes que llegan para cada usuario previamente identificado, el almacenamiento tiene lugar en ordenadores donde residen los buzones.

Cada mensaje tiene conceptualmente tres partes:

- *Sobre (envelope), es la información para transportar el mensaje.*
- *Cabecera (headres), información de control para los clientes (de, para, fecha, asunto, etc)*
- *Cuerpo(body), lo que el usuario desee enviar como contenido.*

Así mismo para enviar un mensaje de correo se deben seguir estos pasos:

- *Escribir el texto del mensaje en un ordenador.*
- *Incluir el identificador de usuario y la dirección de correo electrónica del destinatario.*
- *Enviar el mensaje*

Las direcciones de correo electrónico tienen una estructura típica que es la siguiente:

idusuario@dominio

donde "idusuario" es el identificativo de la persona a la que va dirigido el mensaje, luego viene el símbolo @ que actúa como separador, y a continuación se localiza el buzón de esa persona. Este nombre de ordenador puede constar de varios campos separados por puntos como vemos a continuación en el ejemplo más detallado de la dirección:

pserrano@baustro.fin.ec

3.2.1 Componentes

Hay componentes de correo electrónico que permiten el proceso de trasmitir y administrar mensajes de correo electrónico y garantizar que los mensajes llegan al destino correcto.

MUA (Mail User Agent, Agente de usuario de correo), *Un MUA permite a un usuario, como mínimo, leer y escribir mensajes de correo electrónico, aunque hay otros programas MUA que ofrecen al usuario más funciones, entre las que se incluyen la recuperación de mensajes mediante los protocolos POP e IMAP, la configuración de buzones de correo para almacenar los mensajes o ayuda para presentar los mensajes nuevos a un programa MTA (Mail Transfer Agent, Agente de transferencia de correo) que los enviará al destino final. Un MUA puede ser gráfico, como Mozilla Mail, o pueden tener una interfaz basada en texto sencilla, como Mutt o Pine.*

MTA (Mail Transfer Agent, Agente de transferencia de correo), *Un MTA transfiere los mensajes de correo electrónico entre máquinas que usan el protocolo SMTP. Un mensaje puede pasar por varios MTA hasta llegar al destino final.*

Los agentes MUA de mayores dimensiones y complejidad también sirven para enviar correo. Sin embargo, no se debe confundir esta acción con las funciones propias y verdaderas de estos agentes. Para que los usuarios que no ejecutan un agente MTA propio puedan transmitir los mensajes salientes a una máquina remota para su envío, deben utilizar una capacidad en el MUA capaz de transferir el mensaje a un MTA para el que tengan autorización de uso. Sin embargo, el agente MUA no entrega directamente el mensaje al servidor de correo del destinatario final; esta función está reservada al agente MTA.

MDA (Mail Delivery Agent, Agente de entrega de correo), *Los agentes MTA utilizan programas MDA para entregar el correo electrónico al buzón de un usuario concreto. En muchos casos, el agente MDA es realmente un LDA (Local Delivery Agent, Agente de entrega local), como bin/mail o Procmail. Sin embargo, Sendmail también puede desempeñar la función de un agente MDA, como cuando acepta un mensaje de un usuario local y lo adjunta a su fichero de spool de correo electrónico. Cualquier programa que gestione realmente un mensaje para entregarlo al punto donde lo leerá un agente MUA se puede considerar un agente MDA. aunque no transportan mensajes entre sistemas ni actúan como interfaz para el usuario final. Para enviar y recibir mensajes solo necesita el MTA y MUA, pero el MDA se puede utilizar para ordenar los mensajes antes de que los lea el usuario.*

3.2.2 Protocolos del Correo Electrónico

El correo electrónico de Internet se implementó originalmente como una función del protocolo FTP, se realizaron trabajos con un protocolo que posteriormente se denominaría SMTP ("Simple Mail Transfer Protocol") y hasta hoy en día se sigue utilizando este protocolo, con los avances lógicos que requiere el tipo de transferencia actual, que fue desarrollado pensando en que los sistemas que intercambiarían mensajes, eran grandes computadores, de tiempo compartido y multiusuario conectados permanentemente a la red Internet. Sin embargo, con la aparición de los computadores personales, que tienen una conectividad ocasional, se hizo necesaria una solución para que el correo llegase a estos equipos. Para solventar esta limitación, en 1984 surge POP ("Post Office Protocol"), este protocolo, en su especificación inicial, solo permite funciones básicas como recuperar todos los mensajes, mantenerlos en el servidor y borrarlos. En sucesivas versiones del protocolo (POP2 y POP3) se han ampliado las funciones, permitiendo una mejor gestión del correo.

Estos dos protocolos son los encargados de transportar el correo por toda la red Internet, pero sólo son capaces de transportar mensajes en formato texto ASCII. Para superar esta limitación, se utilizaba hasta hace poco tiempo, programas como UUEncode y UUDecode.

En 1992, surge MIME ("Multipurpose Internet Mail Extensions"), que permite el correo electrónico en otras lenguas que no sea el inglés, además de la transmisión de sonido, gráficos, vídeo, etc. En la actualidad el estándar MIME, es el que se utiliza para la transferencia de datos no tradicionales a través de correo electrónico.

Protocolo Simple de Transmisión de Correo (SMTP)

El Protocolo Simple de Transmisión de Correo ("Simple Mail Transfer Protocol") es el estándar de Internet para el intercambio de correo electrónico entre diferentes ordenadores. SMTP necesita que el sistema de transmisión ponga a su disposición un canal de comunicación fiable y con entrega ordenada de paquetes, con lo cual, el uso del protocolo TCP en la capa de transporte, es lo adecuado, el puerto utilizado por el receptor es el número 25. Para que dos sistemas intercambien correo mediante el protocolo SMTP, no es necesario que exista una conexión interactiva, ya que este protocolo usa métodos de almacenamiento y reenvío de mensajes.

Durante una sesión SMTP el origen y el destino intercambian una secuencia de comandos y respuestas que siguen básicamente los siguientes pasos:

- *Identificación de los hosts*
- *Identificación del remitente del mensaje*
- *Identificación del destinatario del mensaje*
- *Transmisión de los datos (mensaje)*
- *Transmisión de un código que indica el fin de la transacción*

Los códigos de respuesta de SMTP están estructurados de un modo muy similar al FTP, siendo números decimales de tres dígitos e indicando el primero el status del comando y los dos siguientes información más detallada, siendo en general aquellos que comienzan por 1, 2 ó 3 los que indican la realización de un comando con éxito y los que comienzan por 4 ó 5 indican algún tipo de problema, esto consiste en una serie de campos precedidos por unas cabeceras (la mayoría opcionales), seguidas de una línea en blanco y a continuación el texto del mensaje.

Los nombres de campo y su contenido están codificados con caracteres ASCII y existen multitud de cabeceras, las más importantes son las siguientes:

Received:	**Date:**	**From:**	**To:**	**cc:**
Message-Id:	**Reply-To:**	**Sender:**	**Subject:**	**bcc**

Todas las cabeceras deben contener al menos los campos Date, From y To. La mayoría de los programas de correo también crean un identificador del mensaje: Message-Id que se incluye en la cabecera del mensaje, por ejemplo:

Message-Id:<180@gtw_correo>

El identificador está diseñado para ser único en la red; para conseguir este objetivo suele contener además de un número de orden el nombre del host originador del mensaje.

Funcionamiento básico de SMTP

El funcionamiento de SMTP, se inicia con una serie de órdenes y respuestas intercambiadas por el emisor y el receptor, cada orden consta de una línea de texto de cuatro dígitos, las respuestas pueden tener más de una línea y comienzan con un código de tres dígitos ,tomando en cuenta que el primer bit indica la categoría de la respuesta: Respuesta de finalización afirmativa, Respuesta Intermedia Positiva (la orden fue aceptada, pero la acción suspendida), Respuesta de finalización negativa transitoria (no aceptó la orden y la acción no se realizó, es temporal), Respuesta de finalización negativa permanente(no aceptó la orden y la acción no se realizó). Los siguientes cuadros presentan algunas órdenes y respuestas de SMTP.

NOMBRE	FORMATO DE LA ORDEN	DESCRIPCIÓN
HELO	HELO <SP> <dominio><CRLF>	Envía identificación
MAIL	MAIL<SP>FROM:<camino inverso><CRLF>	Identifica origen de correo
RCPT	RCPT <SP>TO:<camino destino><CRLF>	Aborta transacción
DATA	DATA <CRLF>	Transfiere texto msje.
SEND	SEND<SP>FROM:<camino inverso><CRLF>	Envía correo al terminal
QUIT	QUIT <CRLF>	Cierra conexión TCP

Ordenes de SMTP

CODIGO	DESCRIPCIÓN
220	<dominio>Servicio Preparado
221	<dominio>Servicio cerrado el canal de transmisión
354	Comenzar entrada de correo
452	Acción solicitada no ejecutada
551	Usuario no local, por favor intente <camino-destino>
554	Transacción fallida

Respuestas de SMTP

Al establecer la conexión para entregar los mensajes la secuencia es:

- *Emisor establece una conexión TCP con el receptor*
- *Al establecerla, el receptor se identifica así mismo con <220 Service Ready>*
- *El emisor se identifica así mismo con la orden HELO*
- *El receptor acepta la identificación del emisor con <250 OK>*

Una vez establecida la conexión, el emisor envía uno o más mensajes al receptor, con las tres fases lógica siguientes:

- *Una orden MAIL identifica al que originó el mensaje*
- *Una o más órdenes RCPT identifica los destinos de ese mensaje*
- *Una orden DATA que transfiere el texto del mensaje.*

Y para cerrar la conexión, el emisor envía una orden QUIT y espera la respuesta, se inicia una operación de cierre para la conexión TCP y el receptor inicia su cierre luego de enviar su orden QUIT.

Protocolo de Oficina de Correos (POP)

POP es Protocolo de Oficina de Correos ("Post Office Protocol"), se usa para permitir a una estación de trabajo tomar el correo que para ella el servidor tiene almacenado, este protocolo no esta destinado a operaciones extensas sobre el correo, solo, es una vía para recuperarlo, cuando un cliente pide leer su correo el host o cliente que hace uso del servicio establece una conexión de TCP con el host del servidor, cuando esto es realizado el cliente y el servidor intercambian comandos y respuestas hasta que la conexión se cierra, los comandos en el POP3 consisten en una palabra clave misma que es verificada en el servidor para dar acceso a leer los correos.

En la actualidad, se utiliza el protocolo SMTP para el envío de correo y para la recepción de correo se utiliza el protocolo POP, el cual, ya está en su tercera versión desde su aparición (POP3), la cual no posee grandes novedades con

respecto al original, ya que básicamente, sigue permitiendo la descarga de los mensajes llegados a la casilla del usuario.

Funcionamiento de POP

El protocolo POP se basa en el concepto de buzón, que posee un espacio para almacenar los mensajes de correo hasta que se solicite la descarga de estos mensajes. El cliente POP se conecta con el servidor a través del puerto TCP, 110. Para conectarse al servidor, es necesario una cuenta de identificación en dicha máquina (lo que le permite tener un espacio reservado para sus correos). A continuación es necesario verificar que es dueño de la cuenta a través de una clave. Una vez conectado al sistema, el cliente POP puede dialogar con el servidor para saber, en otros, si existen mensajes en la casilla, cuántos mensajes son o para solicitar la descarga de alguno de ellos. Para poder ofrecer estas funciones, el modelo de comunicación POP se basa en estados: estado de autorización, estado de transacción y estado de actualización.

Después de establecer la conexión, el servidor POP se encuentra en un estado de autorización, esperando que el cliente le envíe el nombre y clave de la cuenta de usuario. Cuando se verifica que el nombre y la clave son correctos, el servidor pasa a un estado de transacción. Antes de pasar a este estado, el servidor POP bloquea el buzón para impedir que los usuarios modifiquen o borren el correo antes de pasar al estado siguiente. En este estado de transacción el servidor atiende las peticiones del cliente. Después de enviar al servidor el comando QUIT, el servidor pasa al estado de actualización(estado siguiente). En este estado el servidor elimina los mensajes que están con la marca de borrado y finaliza la conexión.

Los comandos utilizados por el protocolo POP son pocos, el diálogo desde el cliente al servidor, se basa en el envío de comandos, a los que el servidor responde con código y cambiando, cuando corresponda, de un estado a otro, sólo se establecen dos códigos de respuesta, uno para cuando el comando funciona correctamente y otro para cuando no es así. Los códigos de respuesta que el servidor POP envía, van seguidos de una frase que explica o aclara el código, lo que puede ayudar a conocer cual es el motivo de los errores, si se producen. Por ejemplo:

+OK
comando funciona correctamente
+ERR
comando no funciona

Al conectarse al servidor se inicia el estado de autorización, el usuario de correo debe enviar el nombre de la cuenta y la clave para poder continuar, si son correctos, la casilla de esa cuenta pasa a un estado de bloqueo exclusivo, para impedir que los mensajes sean modificados o borrados antes de llegar al estado de actualización del servidor POP. Si no se consigue pasar la casilla al estado de bloqueo exclusivo, se produce un fallo y no se puede pasar al estado de transacción. PASS (Clave) señala al servidor la clave de la cuenta de usuario indicada por el comando USER. Si la

clave no es correcta o la casilla no pasa al estado de bloqueo exclusivo, se produce un error. La sintaxis de este comando es la siguiente: ***PASS clave#13#10***

Cuando el servidor está en estado de autorización y en estado de transacción puede usarse la orden QUIT: ***QUIT#13#10.*** *Y la orden USER le proporciona al servidor el identificador o nombre de la cuenta de usuario.* ***USER id-cuenta#13#10***

En el estado de transacción,, el cliente puede enviar comandos para conocer si tiene o no, correo nuevo, borrar correo (marcar como borrado), recuperarlo, almacenarlo, etc. DELE (Eliminar) marca como eliminado un mensaje con su respectivo número, pero en realidad el servidor no lo elimina hasta que no pasa al estado de actualización, con lo que no se pierde en el caso de que la conexión fallase o que se deseara quitarle la marca de eliminar: ***DELE numero_mensaje#13#10,*** *para recuperara la información del tamaño de los mensaje se utiliza el comando:* ***LIST [numero_mensaje]#13#10;*** *para recuperar o solicitar que el servidor envíe un mensaje determinado el comando es:* ***RETR numero_mensaje#13#10;*** *para anular la marca de borrado de todos los mensajes en la casilla, el comando es:* ***RSET#13#10;*** *para obtener un resumen del contenido de la casilla el comando es:* ***STAT#13#10***

En el estado de actualización no hay comandos, luego de que el estado de transacción envíe al servidor el comando QUIT. En este estado de actualización se eliminan los mensajes que han sido marcados en el estado anterior, se quita el bloqueo exclusivo a la casilla para que pueda actualizarse con nuevo correo y el servidor termina la conexión.

Extensión de Correo de Internet Multipropósito (MIME) y Extensión de Servicios

El primero es Extensión de correo de Internet multipropósito (MIME), permite mensajes para ser declarados como consistentes de datos de 8-bit más que de datos de 7-bit, debido a que los mensajes no pueden ser transmitidos por agentes SMTP puesto que cuando un cliente SMTP intenta enviar datos de 8 bits a un servidor que no soporta esa extensión, el cliente SMTP debe codificar el contenido del mensaje a una representación de 7 bits o retornara un error permanente del servidor.

El protocolo MIME surge por la incapacidad que tiene el SMTP para representar todos los datos que se desean transmitir a través de correo electrónico, en un principio se define el formato normal de mensajes textuales en Internet (define sintaxis de cuerpo y cabecera), pero su crecimiento va más allá de los limites de Internet y más que el transporte por Internet SMTP, . Cuando el formato se ha usado en forma más extensa, al considerar mensajes multimediales que incluyan audio o imágenes. Incluso, en el caso de texto, SMTP es inadecuado para las necesidades de usuarios, cuyos idiomas requieren el uso de un conjunto de caracteres más rico que EE.UU.-ASCII de 7 bits.

Las especificaciones de MIME incluyen los siguientes elementos:

- *Define cinco nuevos campos para la cabecera, que proporcionan información sobre el cuerpo del mensaje.*
- *Define varios formatos de contenido, que puedan soportar correo electrónico multimedia.*
- *Define esquemas de codificación de transferencia, posibilitando la conversión del contenido a cualquier formato.*

Los campos dados en la cabecera son:

- *Versión MIME, versión que debe ser 1.0.*
- *Tipo de Contenido, describe los datos del contenido con suficiente detalle que el agente usuario receptor pueda escoger el mecanismo para tratar el contenido.*
- *Codificación de transferencia del contenido, indica el tipo de transformación que se ha utilizado para representar el cuerpo del mensaje.*
- *Descripción del contenido, descripción del texto del objeto que acompaña al cuerpo.*

Los tipos de contenido, es lo particular o especial, por ello se requiere de software especial para obtener el significado completo del texto.

TIPO	SUBTIPO	DESCRIPCIÓN
Texto	Nativo	vía identificación
	Mezclado	Identifica origen de correo
	Alternativo	Aborta transacción
Mensaje	Parcial	Usado para permitir fragmentación de correos grandes
	Cuerpo-externo	Contiene un puntero a un objeto que existe en otro lugar
Imagen	jpeg	Imagen en formato JPEG, codificación JFIF
	gif	Imagen en formato GIF
Video	mpeg	Formato MPEG
Audio	Básico	Codificación en ley-mu de canal único de 8 bits
Aplicación	postscript	Postscript de adobe

3.2.3 Correo WEB

Existe la posibilidad de consultar nuestro correo desde cualquier navegador Web, estemos donde estemos. Para ello basta con teclear la dirección e introducir nuestro nombre de usuario y contraseña, además se necesita un navegador que permita conexiones seguras (https). Prácticamente todos los que se utilizan actualmente lo soportan (IExplorer, Netscape, Opera, ...). Un ejemplo de dirección puede ser:

http://www.bcoazuay.es/correo/webmail/

Características

- *El correo electrónico es el enlace de comunicación entre una organización específica y el resto de la comunidad.*
- *La capacidad permitida para guardar mensajes en el servidor se puede definir.*
- *Se puede consultar desde cualquier parte del mundo con una computadora que tenga acceso a Internet.*
- *Se puede facilitar información que se registra en el servidor, para que pueda ser actualizada o ingresada por los usuarios, como un directorio que guarda datos completos los cuales incluyen teléfonos, correo electrónico y direcciones postales, entre otros.*
- *Acceso a través de una página web y/o por medio de un programa de correo local (vía POP) y al utilizar el correo vía POP el disco duro es el único límite de espacio.*
- *El correo vía WEB funciona como una página de Internet, con acceso directo en la dirección, al teclear la clave de usuario y la contraseña el acceso será inmediato, mientras que el correo local solo puede ser revisado cuando se encuentre dentro del dominio de su red.*
- *Cuando se utiliza un software de correo electrónico como Netscape Messenger o Outlook, utiliza el protocolo POP (POST OFFICE PROTOCOL) en realidad está bajando la información a la máquina que estás usando, por eso se dice que trabaja de manera local, mientras que con el webmail, la información se queda guardada en el servidor de correo.*

3.2.4 Caso Práctico

SMTP como servicio usa el puerto 25. La idea es interactuar a través de comandos con el MTA usando un telnet simple.

Conectarse al smtp server:

- ***Host Name*** *smtp server (sigma.eafit.edu.co)*
- ***Puerto*** *25*
- ***Term Tyme:*** *Ansy*

Se ejecuta el comando telnet de la siguiente forma: ***telnet sigma.eafit.edu.co 25***

El servidor debe contestar de la siguiente forma:

220 smtp5.jps.net ESMTP Sendmail 8.9.3/8.9.0; Wed, 29 Dec 1999 14:10:55 -0800 (PST)

El comando 220 que devuelve el servidor, indica que el servicio esta listo y puede continuar comunicándose. Para continuar se escribe el siguiente comando:

helo me

El servidor le contesta con los siguientes comandos:

250-smtp5.jps.net Hello 216-119-33-50.o1.jps.net [216.119.33.50], pleased to meet you
250-8BITMIME
250-SIZE 20000000
250-DSN
250-ONEX
250-XUSR
250 HELP

El commando 250 que devuelve el servidor, indica que se ha completado la acción correctamente. A continuación el siguiente comando:

mail from: almontoy@sigma.eafit.edu.co

El servidor le devolverá algo como:

250 youraddress@yourdomain.com... Sender ok

Esta línea indica el formato correcto para enviar la dirección de correo electrónico ha sido enviado. El siguiente comando:

rcpt to: almontoy@eafit.edu.co

La respuesta es: ***250 goingto@domain.com... Recipient ok***

Al ingresar el comando RCPT, indico que la comunicación a esa dirección queda anulada, el servidor me responde el formato correcto de la dirección. Ahora se escribe:

data

Respuesta: ***354 Enter mail, end with "." on a line by itself***

Me indica que el inicio del mensaje va con una "," y el final con un ".".

Ahora las siguientes líneas:

To: Alcides <almontoy@sigma.eafit.edu.co>
From: Your Name <youraddress@yourdomain.com>
Subject: The subject of the message
Your message should go here.
Type what ever you want, it doens't really make a difference.

Se debe finalizar el mensaje colocando un punto al final de la linea y enter. Debe recibirse este comando:

250 OAA24319 Message accepted for delivery

El comando 250 me indica que el mensaje ha sigo aceptado. Para finalizar se escribe Quit y enter,

3.3 Servicios de Información: WWW

Se denomina World Wide Web, al formado de toda la información disponible en Internet que puede ser vista a través de un cliente Web, debe hacerse una distinción clara entre Internet y el WWW. Este último es uno más de los servicios de información disponibles en Internet, una red de ordenadores que se comunican a través del protocolo TCP/IP. La evolución de los clientes Web como interfaz casi única de acceso a numerosos servicios de Internet contribuye al equívoco que supone utilizar indistintamente ambos términos.

El proyecto Web ha basado su éxito en un diseño muy acertado de todos sus componentes, que, a partir de su relativa simplicidad, permite la construcción de sofisticados sistemas de información, basado en un modelo cliente-servidor estricto, en el que los intercambios de información entre clientes y servidores se realizan a través de sencillas peticiones.

Los servidores HTTP son el núcleo del sistema de distribución de información, se encuentra la copia original de los documentos a distribuir, y los clientes tienen capacidad de recogerlos. Son capaces de manejar información, multimedia, apoyados en el estándar MIME, acceder a un documento a través de su ***URL*** *(Universal Resource Locator), que permite asignar una dirección a casi cualquier recurso disponible en Internet. Otras características de World Wide Web son:*

- *Es muy fácil de utilizar las interfaces de los clientes Web.*
- *"Todo lo que se puede saber sobre el Web está en el propio Web". Esta es una frase que resume la filosofía de desarrollo del proyecto WWW. A través de un cliente Web se puede acceder a ayudas sobre su manejo, aprender más sobre el Web, conocer cómo desarrollar nuevas páginas.*
- *Por su flexibilidad, puede dar soporte a multitud de servicios diferentes, por lo que resulta interesante para empresas, centros de enseñanza, se puede acceder a servicios comerciales y publicitarios, cursos y guías, bases de datos, etc.*
- *Es muy fácil publicar nueva información, así como incorporar información en formato electrónico de la que se disponía previamente.*
- *Está en continua evolución, y cada día sus capacidades de acceso y representación de información se vuelven más sofisticadas.*
- *Ha sido la principal causa del espectacular crecimiento que Internet ha tenido en los últimos cuatro años, tanto en el número de usuarios como en el volumen de información disponible.*
- *También sirve como soporte para las denominadas "Intranets", que son versiones reducidas de Internet para empresas, grupos de trabajo, etc., que utilizan la tecnología de publicación del Web, para intercambiar información entre grupos de trabajo.*

3.3.1 Elementos

El WWW se sustenta en cinco elementos fundamentales:

- ***Un** protocolo **de comunicación llamado** **HTTP** **(HyperText Transfer Protocol), es un protocolo de transferencia de hipertextos, audio, imágenes o cualquier información accesible a través del internet.***

- ***Un lenguaje para escribir documentos hipermedia (**HTML** ó HyperText Markup Language), es u**tilizado para crear y reconocer documentos hipermedia, permite separar la presentación del contenido.

  *Un documento HTML es un conjunto de caracteres ASCII de 7 bits, con códigos para: Estilos del texto, Títulos de documentos, secciones, Párrafos, Listas, Hiperenlaces, Formularios. Los documentos HTML suelen tener la extensión **html** o **htm**, si bien cada servidor HTTP puede configurar, a través de la base de datos de MIME, las extensiones asignadas (puede haber varias simultáneamente). Un caso muy común son los documentos .asp, del Active Server Pages de Microsoft.*

- ***Un sistema para designar el lugar donde se encuentra el documento en la Red llamada **URL** (Uniform Resource Locator),*** URL especifica el método de acceso (protocolo) en general http. La dirección de la computadora en la que reside el servicio (opcionalmente puede llevar un puerto: como se ve en el ejemplo:

  http://gsyc.escet.urjc.es**:80/**ficheros/fichero1.html

  El resto de la URL especifica el camino y el nombre del fichero, opcionalmente puede llevar un nombre de sección, separado por #:

  ***http://gsyc.escet.urjc.es/ficheros/fichero1.html**#secc2*

  Las computadoras que ofrecen un servicio de WWW suelen nombrarse con www al principio de su dirección: `www.baustro.fin.ec`. Las URLs no se limitan al entorno Web, sino que permiten localizar información en otros servicios Internet: Gopher, FTP, News, Telnet, Wais,…, y son un elemento fundamental para el funcionamiento del Web, ya que permiten que los clientes Web manejen un único formato de direcciones, independientemente del tipo y situación del recurso que se solicita. Las URLs definen, a través de un sencillo formato, la ubicación en la red de la información deseada. La estructura de una URL aparece en la siguiente tabla (las partes con líneas continuas son obligatorias, mientras que las otras son opcionales):

Protocolo	:/ /	Dirección del servidor	:puerto	/	Situación del recurso

Protocolo : nombre del servicio de información al que debe acceder el cliente Web.

Dirección del servidor: dirección IP o nombre DNS del servidor. Es el ordenador con cuyo servicio (http, ftp, gopher,...) se establece la comunicación.

Puerto : puerto local dentro de la máquina remota. No siempre se utiliza; de esta forma se puede alterar el valor por defecto asociado a cada servicio de información (80 para http, 23 para telnet,…).

Situación: permite añadir información opcional, característica del protocolo, que se corresponde a la dirección del objeto requerido dentro de la estructura de información del propio servidor que lo ofrece.

Otros ejemplos:

***file://gsyc.escet.urjc.es/pub/sonido.au** Trae y emite el sonido*

***file://gsyc.escet.urjc.es/imagen.gif** Trae y muestra la imagen*

***file://gsyc.escet.urjc.es/pub/** Contenido del directorio*

***http://gsyc.escet.urjc.es/~vmo/index.html** Se conecta a un servidor HTTP y trae un fichero HTML*

***ftp://www.xerox.com/pub/file.txt** Abre una sesión FTP con www.xerox.com y trae un fichero de texto*

***news:urjc.csalud.fisio** Lee las novedades*

***mailto:vmo@gsyc.escet.urjc.es** Envía correo electrónico*

- *Un conjunto de aplicaciones o programas llamados browsers o navegadores y los servidores **httpd**, los cuales se dividen el trabajo de presentar y servir la información multimedia al usuario. Por ejemplo el Apache es un servidor web, que permite el alojamiento de páginas web en una máquina específica, esta herramienta tiene varias funciones tales como: permitir a los usuarios tener sus propias páginas web, restricción a determinados sitios web, conexiones seguras a través de SSL, configuración de módulos de programación. El archivo de configuración del servidor Web Apache es /etc/httpd/conf/httpd.conf. El archivo httpd.conf está bien comentado y es bastante autoexplicativo. La configuración predeterminada de secure Web server funciona para los ordenadores de la mayoría de los usuarios, así que probablemente no necesitará cambiar ninguna de las directivas en el fichero httpd.conf.*

- ***Documentos Hipermedia**, que son objetos conectados a través de enlaces (Hiperlinks).* A los documentos del WWW se les conoce como documentos hipermedia. Se puede definir hipertexto como una red de información textual de naturaleza no secuencial. Un libro presenta la información de forma secuencial, es decir primero hay que leer la página 1 después la 2, y así sucesivamente. Si concibiéramos un libro como un hipertexto no necesariamente tendríamos que seguir ese orden, por ejemplo: de la página 1 podríamos pasar a la 123 y de ésta a la 12 o seguir el orden secuencial expuesto más arriba. En todo hipertexto existe tres elementos fundamentales:

    - *Nodo: Es la unidad básica de información del hipertexto, por ejemplo una página de texto.*
    - *Enlace: Es una conexión existente entre dos nodos de manera que a través del mismo se puede pasar de un nodo a otro.*
    - *Anclaje: Es la zona del nodo donde se inicia el enlace hacia otro nodo.*

Por lo tanto se puede definir hipermedia como una red de información cuyos documentos que la constituyen integran texto, imágenes, sonido y video (hipertexto + multimedia), conociéndose a cada uno de los documentos como documentos hipermedia.

3.3.2 Protocolo http

HTTP fue inventado para que las computadoras se comuniquen mientras intercambian documentos, agregando conectividad e interfaces. Usando HTTP, una computadora que pida un archivo a otra sabrá, al recibirlo, si se trata de imagen, video o texto. HTTP es un protocolo sin estados, cada transacción se trata independientemente, así es que una sesión normal de un usuario con el cliente Web supone obtener una secuencia de páginas y documentos web, la secuencia se obtiene rápidamente y las localizaciones de las distintas páginas y documentos puede ser una serie de servidores distribuidos mundialmente.

Según la especificación del protocolo, "HTTP es un protocolo del nivel de aplicación con la agilidad y velocidad necesaria para sistemas de información distribuidos, colaborativos y de hipermedia. Es un protocolo orientado a objetos, genérico, que puede usarse para muchas tareas extendiendo sus métodos.

Propiedades de HTTP

Esquema de direccionamiento integral: *El protocolo usa el concepto de referencia dado por URI (Universal Resource Identifier) como una ubicación o como un nombre - URL y URN respectivamente -para indicar la fuente donde debe aplicarse un método. Cuando un hiperlink HTML se conforma, la URL es de la forma http://host:número_puerto/path/archivo.html. Para decir algo más general, la referencia URL es del tipo servicio://host/archivo.extensión. De esta manera el protocolo puede abarcar los servicios de Internet más básicos.*

HTTP también se usa para la comunicación entre agentes y gateways, permitiendo el acceso a otros protocolos de Internet existentes, como SMTP, NNTP, FTP, Gopher, etc. HTTP está diseñado para permitir la comunicación con esos gateways, vías servidores proxy, sin pérdida de la información transportada por estos otros protocolos.

Arquitectura Cliente-Servidor: *HTTP se basa en el paradigma pedido/respuesta. La comunicación generalmente se lleva a cabo sobre una conexión TCP/IP en Internet. El puerto por defecto es el 80, pero otros puertos también pueden usarse. Esto no imposibilita que el protocolo se implemente arriba de algún otro protocolo de internet, siempre y cuando se garantice la confiabilidad.*

Un programa cliente establece la conexión con un programa del servidor y envía un pedido a este último mediante el método request, URI y versión de protocolo, seguido por un mensaje que contiene los modificadores del pedido, información sobre el cliente y contenido. El servidor responde con una línea de estado, incluyendo su versión de protocolo y un mensaje de éxito o error, seguido de un mensaje que contiene información del servidor, meta-información y contenido.

Sin conexión: *Aunque dijimos que el cliente establece una conexión con el servidor, se dice que el protocolo es sin conexión porque una vez que el pedido está satisfecho, la conexión cae. Otros protocolos mantienen la conexión abierta. Por ejemplo, en una sesión FTP puedes moverte de un directorio remoto a otro y el servidor irá siguiendo tus movimientos, para saber en todo momento quién eres y dónde estás.*

Mientras que esto simplifica enormemente la construcción del servidor y libera al mismo de carga que disminuiría la performance, el seguimiento del usuario se hace imposible. Además, se debe recurrir a múltiples conexiones para documentos que consisten de más de una imagen en línea, ya que cada una se recupera individualmente en una conexión.

Sin estado: *Luego que el servidor responde a un pedido del cliente, la conexión cae y es olvidada. Es decir, no hay una memoria de conexiones de un cliente. Las implementaciones de servidor HTTP puras tratan todos los pedidos como si fueran únicos y nuevos.*

Las aplicaciones CGI pueden salvar esto codificando el estado (o un identificador de estado) en campos ocultos, en la información del path o en URLs que se

retornan al navegador. Los dos primeros métodos devuelven el estado (o su identificador) cuando el formulario es enviado de vuelta al servidor; el método que usa la URL sólo devuelve el estado (o su identificador) si el usuario hace click en el link y el link vuelve al servidor.

Es aconsejable no codificar todo el estado, sino almacenarlo en un archivo e identificarlo con un identificador único, como un entero secuencial. Los programas contadores de visitantes pueden adaptarse muy bien a esto, por lo que son muy útiles. Entonces sólo debes enviar el identificador de estado en el formulario, lo que ahorra tráfico de red. Sin embargo, debes acordarte de hacer el mantenimiento de los archivos de estado.

***Mensajes**, son en una petición de un cliente al servidor y en la respuesta del servidor al cliente.*

Las peticiones y respuestas pueden ser simples o completas. La diferencia es que en las peticiones y respuestas completas se envían cabeceras y un contenido. Este contenido se pone después de las cabeceras dejando una línea vacía entre las cabeceras y el contenido. En el caso de peticiones simples, sólo se puede usar el método GET y no hay contenido. Si se trata de una respuesta simple, entonces ésta sólo consta de contenido. Esta diferenciación entre simples y completas se tiene para que el protocolo HTTP/1.0 pueda atender peticiones y enviar respuestas del protocolo HTTP/0.9.

Petición,** de un cliente a un servidor ha de incluir el método que se aplica al recurso, el identificador del recurso y la versión del protocolo que usa para realizar la petición. Para mantener la compatibilidad con el protocolo HTTP/0.9 se permite una petición simple con el formato: **`GET'' SP URI CRLF

Donde SP es un espacio, URI es la URI del recurso al que hace referencia la petición y CRLF es un retorno de carro y nueva línea.

En el caso de que la petición se haga con el protocolo HTTP/1.0 o con el protocolo HTTP/1.1 la petición sigue el formato:

***La línea de petición,** comienza indicando el método, seguido de la URI de la petición y la versión del protocolo, finalizando la línea con CRLF:*

En el caso del protocolo HTTP/0.9 sólo se permite el método GET, con el protocolo HTTP/1.0 GET, POST y HEAD y con el protocolo HTTP/1.1 OPTIONS, GET, HEAD, POST, PUT, DELETE y TRACE. En caso de que un servidor tenga implementado un método, pero no está permitido para el recurso que se pide, entonces ha de devolver un código de estado 405 (método no permitido). Si lo que ocurre es que no tiene implementado el método, entonces devuelve un código 501 (no implementado). Los únicos métodos que deben soportar los servidores de forma obligatoria son los métodos GET y HEAD.

En el apartado de cabeceras, éstas pueden ser de tres tipos: ***cabeceras generales, de petición y de entidad.*** *Las cabeceras generales son las que se aplican tanto a peticiones como a respuestas, pero no al contenido que se transmite. Las cabeceras de petición permiten al cliente pasar información al servidor sobre la petición y sobre el cliente. Las cabeceras de entidad permiten definir información adicional sobre el contenido que se transmite y en caso de que no haya contenido, sobre el recurso al que se quiere acceder con la petición. El contenido (si está presente) está en un formato con una codificación definida en las cabeceras de entidad.*

Respuesta *, Después de recibir e interpretar una petición, el servidor debe responder con un mensaje HTTP. Este mensaje tiene el siguiente formato:*

La línea de estado, *es la primera línea de la respuesta y consiste en la versión de protocolo que se utiliza, seguida de una indicación de estado numérica a la que puede ir asociada una frase explicativa. El formato es el siguiente:*

Versión del protocolo SP Código de estado SP Frase explicativa CRLF

El ***código de estado*** *es un número de 3 dígitos que indica si la petición ha sido atendida satisfactoriamente o no, y en caso de no haber sido atendida, indica la causa. Los códigos se dividen en cinco clases definidas por el primer dígito del código de estado.*

- 1xx: Informativo. La petición se recibe y sigue el proceso. Esta familia de respuestas indican una respuesta provisional. Este tipo de respuesta está formada por la línea de estado y las cabeceras. Un servidor envía este tipo de respuesta en casos experimentales.
- 2xx: Éxito. La acción requerida por la petición ha sido recibida, entendida y aceptada.
- 3xx: Redirección. Para completar la petición se han de tomar más acciones.
- 4xx: Error del cliente. La petición no es sintácticamente correcta y no se puede llevar a cabo.
- 5xx: Error del servidor. El servidor falla al atender la petición que aparentemente es correcta.

Algunos de los códigos más comúnmente usados y las frases asociadas son:

- 100, continuar.
- 101, cambio de protocolo.
- 200, éxito.
- 201, creado.
- 202, aceptado.
- 203, información no autoritativa.
- 204, sin contenido.
- 302, movido temporalmente.

- 303, ver otros.
- 304, no modificado.
- 305, usar *proxy*.
- 400, petición errónea.
- 401, no autorizado.
- 406, no se puede aceptar.
- 407, se requiere autentificación *proxy*.
- 408, límite de tiempo de la petición.
- 500, error interno del servidor.
- 501, no implementado.
- 502, puerta de enlace errónea.
- 503, servicio no disponible.
- 505, versión de protocolo HTTP no soportada.

*La **frase explicativa**, es eso, una frase corta que explica el código de estado enviado al cliente. Se pueden usar más códigos, pero las aplicaciones HTTP no tienen que conocer todos los códigos definidos y su significado, pero sí están obligadas a conocer su clase y tratar los códigos desconocidos como el primer código de la clase (x00). Además, las **cabeceras de la respuesta**, son de tres tipos: las cabeceras generales, las cabeceras de la respuesta y las cabeceras de entidad.*

Las cabeceras de respuesta permiten al servidor enviar información adicional al cliente sobre la respuesta. Estos campos dan información sobre el servidor y acceso al recurso pedido.

***METODOS,** Un método se dice que es seguro si no provocan ninguna otra acción que no sea la de devolver algo (no produce efectos laterales). Estos métodos son el método GET y el método HEAD. Para realizar acciones inseguras (las que afectan a otras acciones) se pueden usar los métodos POST, PUT y DELETE. Aunque esto está definido así, no se puede asegurar que un método seguro no produzca efectos laterales, porque depende de la implementación del servidor.*

Un método es ídem potente si los efectos laterales para N peticiones son los mismos que para una sola petición. Los métodos ídem potentes son los métodos GET, HEAD, PUT y DELETE.

***Método OPTIONS,** Este método representa un petición de información sobre las opciones de comunicación disponibles en la cadena petición-respuesta identificada por la URI de la petición. Esto permite al cliente conocer las opciones y requisitos asociados con un recurso o las capacidades del servidor.*

La respuesta sólo debe incluir información sobre las opciones de comunicación. Si la URI es ``'', entonces la petición se aplica al servidor como un conjunto. Es decir, contesta características opcionales definidas por el servidor, extensiones del protocolo.*

***Método GET**, El método GET requiere la devolución de información al cliente identificada por la URI. Si la URI se refiere a un proceso que produce información, se devuelve la información y no la fuente del proceso.*

El método GET pasa a ser un GET condicional si la petición incluye las cabeceras If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match o If-Range. Estas cabeceras hacen que el contenido de la respuesta se transmita sólo si se cumplen unas condiciones determinadas por esas cabeceras. Esto se hico para reducir el tráfico en las redes.

También hay un método GET parcial, con el que se envía sólo parte del contenido del recurso requerido. Esto ocurre cuando la petición tiene una cabecera Range. Al igual que el método GET condicional, el método GET parcial se creó para reducir el tráfico en la red.

***Método HEAD**, El método HEAD es igual que el método GET, salvo que el servidor no tiene que devolver el contenido, sólo las cabeceras. Estas cabeceras que se devuelven en el método HEAD deberían ser las mismas que las que se devolverían si fuese una petición GET.*

Este método se puede usar para obtener información sobre el contenido que se va a devolver en respuesta a la petición. Se suele usar también para chequear la validez de links, accesibilidad y modificaciones recientes.

***Método POST**, El método POST se usa para hacer peticiones en las que el servidor destino acepta el contenido de la petición como un nuevo subordinado del recurso pedido. El método POST se creó para cubrir funciones como la de enviar un mensaje a grupos de usuarios, dar un bloque de datos como resultado de un formulario a un proceso de datos, añadir nuevos datos a una base de datos, etc..*

La función llevada a cabo por el método POST está determinada por el servidor y suele depender de la URI de la petición. El resultado de la acción realizada por el método POST puede ser un recurso que no sea identificable mediante una URI.

***Método PUT**, El método PUT permite guardar el contenido de la petición en el servidor bajo la URI de la petición. Si esta URI ya existe, entonces el servidor considera que esta petición proporciona una versión actualizada del recurso. Si la URI indicada no existe y es válida para definir un nuevo recurso, el servidor puede crear el recurso con esa URI. Si se crea un nuevo recurso, debe responder con un código 201 (creado), si se modifica se contesta con un código 200 (OK) o 204 (sin contenido). En caso de que no se pueda crear el recurso se devuelve un mensaje con el código de error apropiado.*

La principal diferencia entre POST y PUT se encuentra en el significado de la URI. En el caso del método POST, la URI identifica el recurso que va a manejar en contenido, mientras que en el PUT identifica el contenido. Un recurso puede tener distintas URI.

***Método DELETE**, Este método se usa para que el servidor borre el recurso indicado por la URI de la petición. No se garantiza al cliente que la operación se lleve a cabo aunque la respuesta sea satisfactoria.*

***Método TRACE**, Este método se usa para saber si existe el receptor del mensaje y usar la información para hacer un diagnóstico. En las cabeceras el campo Via sirve para obtener la ruta que sigue el mensaje. Mediante el campo Max-Forwards se limita el número de pasos intermedios que puede tomar. Esto es útil para evitar bucles entre los proxy. La petición con el método TRACE no tiene contenido.*

***Cabeceras generales**, Los campos de este tipo de cabeceras se aplican tanto a las peticiones como a las respuestas, pero no al contenido de los mensajes. Estas cabeceras son:*

- *Cache-Control, son directivas que se han de tener en cuenta a la hora de mantener el contenido en una caché.*
- *Connection, permite especificar opciones requeridas para una conexión.*
- *Date, representa la fecha y la hora a la que se creó el mensaje.*
- *Pragma, usado para incluir directivas de implementación.*
- *Transfer-Encoding, indica la codificación aplicada al contenido.*
- *Upgrade, permite al cliente especificar protocolos que soporta.*
- *Via, usado por pasarelas y proxies para indicar los pasos seguidos.*

***Cabeceras de petición,** Este tipo de cabeceras permite al cliente pasar información adicional al servidor sobre la petición y el propio cliente. Estas cabeceras son las siguientes:*

- *Accept, indican el tipo de respuesta que acepta.*
- *Accept-Charset, indica los conjuntos de caracteres que acepta.*
- *Accept-Encoding, que tipo de codificación acepta.*
- *Accept-Language, tipo de lenguaje de la respuesta que se prefiere.*
- *Authorization, el agente de usuario quiere autentificarse con el servidor.*
- *From, contiene la dirección de correo que controla en agente de usuario.*
- *Host, especifica la máquina y el puerto del recurso pedido.*
- *If-Modified-Since, para el GET condicional.*
- *If-Match, para el GET condicional.*
- *If-None-Match, para el GET condicional.*
- *If-Range, para el GET condicional.*
- *If-Unmodified-Since, para el GET condicional.*
- *Max-Forwards, indica el máximo número de elementos por los que pasa.*
- *Proxy-Authorization, permite que el cliente se identifique a un proxy.*
- *Range, establece un rango de bytes del contenido.*
- *Referer, indica la dirección donde obtuvo la URI de la petición.*

- *User-Agent, información sobre el agente que genera la petición.*

***Cabeceras de respuesta**, Permiten al servidor pasar información adicional al cliente sobre la respuesta, el propio servidor y el recurso solicitado. Son los campos:*

- *Age, estimación del tiempo transcurrido desde que se creó la respuesta.*
- *Location, se usa par a redirigir la petición a otra URI.*
- *Proxy-Authenticate, ante una respuesta con el código 407 (autentificación proxy requerida), indica el esquema de autentificación.*
- *Public, da la lista de métodos soportados por el servidor.*
- *Retry-After, ante un servicio no disponible da una fecha para volver a intentarlo.*
- *Server, información sobre el servidor que maneja las peticiones.*
- *Vary, indica que hay varias respuestas y el servidor ha escogido una.*
- *Warning, usada para aportar información adicional sobre el estado de la respuesta.*
- *WWW-Authenticate, indica el esquema de autentificación y los parámetros aplicables a la URI.*

***Cabeceras de entidad,** Como su nombre indica, los campos de este tipo aportan información sobre el contenido del mensaje o si no hay contenido, sobre el recurso al que hace referencia la URI de la petición. Los campos de este tipo son:*

- *Allow, da los métodos soportados por el recurso designado por la URI.*
- *Content-Base, indica la URI base para resolver las URI relativas.*
- *Content-Encoding, indica una codificación adicional aplicada al contenido (a parte de la aplicada por el tipo).*
- *Content-Language, describe el idioma del contenido.*
- *Content-Length, indica el tamaño del contenido del mensaje.*
- *Content-Location, da información sobre la localización del recurso que da el contenido del mensaje.*
- *Content-MD5, es un resumen en formato MD5 (RFC 1864) para chequear la integridad del contenido.*
- *Content-Range, en un GET parcial, indica la posición del contenido.*
- *Content-Type, indica el tipo de contenido que es.*
- *Etag, define una marca para el contenido asociado.*
- *Expires, indica la fecha a partir de la cual la respuesta deja de ser válida.*
- *Last-Modified, indica la fecha de la última modificación.*

3.3.3 Servidor WEB

El Servidor Web no está simplemente manejando archivos sino que también está procesando información generando una página dinámica. En casi todos los casos, el servidor Web utiliza algo llamado "Scripts CGI" para realizar esta magia (Páginas Web).

Se puede ver de esta descripción que un servidor Web puede ser una pieza simple de software. Sólo toma el archivo especificado con el comando GET, y lo envía al servidor. Incluso usted puede crear su propio código para generar su propio servidor Web con alrededor de 500 líneas de código en un lenguaje de programación como el C. Obviamente, un servidor de nivel empresarial es muy diferente, pero los principios básicos son los mismos.

La mayoría de servidores añaden algún nivel de seguridad a sus tareas. Por ejemplo, si usted ha ido a alguna página y el navegador presenta una ventana de diálogo que pregunta su nombre de usuario y contraseña, ha encontrado una página protegida por contraseñas. El servidor deja que el dueño o el administrador del servidor mantenga una lista de nombres y contraseñas para las personas a las que se les permite ver la página, y el servidor deja que sólo esas personas quienes saben la contraseña tengan acceso. Los servidores más avanzados añaden seguridad para permitir una conexión encriptada enter el servidor y el navegador para que información de suma importancia como números de tarjetas de crédito puedan ser enviados por internet. Esto es prácticamente lo que hace un servidor web que "entrega" páginas. Pero también existen las llamadas "Páginas Web Dinámicas" que permiten:

- *Cualquier libro de invitados le permite ingresar un mensaje en un formulario HTML y entonces, la próxima vez que el libro es visto, la página tendrá la nueva entrada.*
- *La forma WhoIs en InterNic le permite registrar un dominio en un formulario, y la página regresada es diferente dependiendo del nombre del dominio ingresado.*
- *Cualquier máquina de búsqueda le permite ingresar texto en un formulario HTML, y entonces, dinámicamente crea una página basada en el texto ingresado.*

En todos estos casos, el servidor Web no está simplemente manejando archivos. Está procesando información y generando una pagina basándose en el interrogante. En casi todos los casos, el servidor Web utiliza algo llamado "Scripts CGI" para realizar esta magia

3.3.3.1 Autenticación

Es importante el crecimiento de la publicación de grandes volúmenes de información a través de Internet que constituye un medio conveniente para acceder a esa información de una manera ágil y eficaz, contando además con

la importante grado de una disponibilidad global, la posibilidad de crear un canal de comunicaciones bidireccional, gracias al cual los usuarios no sólo son capaces de recuperar información de un servidor web, sino también de transmitírsela, principalmente a través de formularios, representa una forma igualmente eficiente de suministrar datos personales e información privada desde cualquier lugar del mundo.

Aunque la información sea fácilmente distribuida, nunca debería suministrarse información confidencial por Internet ni almacenarse en servidores web sin ningún tipo de protección, especialmente en lo que se refiere a datos financieros y comerciales sensibles. A medida que crece la cantidad de información públicamente disponible y transportada a través de Internet, también lo hace la necesidad de asegurarla en parte o en su totalidad, protegiéndola de ojos indiscretos, pero no en detrimento de su facilidad de acceso.

Se trata por tanto de alcanzar un compromiso en el acceso a la información y su seguridad, compromiso que proteja la confidencialidad e integridad, tanto de los datos almacenados en el servidor, como de los que están siendo transportados hacia/desde el servidor; se deben presentar técnicas más ampliamente utilizadas para proteger ciertas áreas del servidor web en plataformas Windows NT y Linux, se muestra de forma sencilla cómo se pueden crear grupos autorizados de acceso, sirviéndose de distintos criterios: dirección IP o nombre de host, autenticación básica, certificados digitales, etc. Asimismo, se debe implantar la solución SSL para proteger el tráfico del servidor, de manera que tanto los datos confidenciales que viajan desde el servidor al usuario, como en sentido inverso, resulten protegidos durante su transporte a través de redes telemáticas.

3.3.3.2 Cookies

Los cookies son pequeños archivos enviados al disco duro de su PC por buena parte de los sitios web que visita. Estos archivos tienen mala fama en Internet, pero en realidad son inofensivos, útiles y se pueden controlar.

Las cookies son archivos de texto que le permiten a un sitio web conocer sus preferencias: qué paginas del sitio suele visitar, cuál es su nombre (si usted lo ha suministrado), conque frecuencia visita ese sitio, etc.

Una cookie, es una estructura: ***Nombre*** *Contiene un identificador único de la cookie, actualmente puede ser cualquier string dado que es empaquetado de acuerdo al www-encoding usual.*

Valor *Un string que contiene el valor asociado al Nombre de la cookie.*

Fecha expiración *La fecha en que la cookie será borrada del disco. Las fechas en JavaScript se manejan como enteros que contienen el número de milisegundos transcurridos desde la medianoche del 1 de enero de 1970. Si es NULL se entiende que la cookie se borr´ al terminar la sesión*

Ruta de validez *A partir de cuales directorios del servidor que originó el requerimiento de grabar-cookie será válida la cookie. Si es NULL se entiende que su valor es: /, o sea, válida en todo el servidor.*

Dominio de validez *Indica bajo que dominio será válida la cookie.*

.Podemos almacenar información anónima a través del uso de varias tecnologías, entre ellas las "cookies". Una cookie es un dato que un sitio web puede enviar a su navegador, que podrá ser almacenada en su sistema. Algunas páginas usan cookies, u otras tecnologías para darle un mejor servicio cuando vuelva al sitio web. Puede configurar su navegador para que le notifique antes de recibir una cookie, dándole la oportunidad de decidir si quiere o no aceptarla. Usted puede también configurarlo para que las cookies queden desactivadas. Sin embargo, si lo hace, algunas áreas de determinados sitios web podrían no funcionar adecuadamente

Si desactiva las cookies, las tecnologías de ayuda web todavía detectarán visitas anónimas a estas páginas, pero los avisos que se generen no podrán ser asociados con cualquier otra información de cookies anónima y serán ignoradas.

Funciones

GetCookie, *Retorna el valor de la cookie especificada por name. name String que contiene el nombre de la cookie.*

SetCookie, *Crea o actualiza una cookie. name String que contiene el nombre de la cookie. value String que contiene el valor de la cookie. Puede ser cualquier string de caracteres validos. expires (opcional) Objeto de tipo Date que contiene la fecha de expiración de la cookie. Si se omite o es null, la cookie se borrara al finalizar la sesión. path (opcional) String que indica la ruta para la cual la cookie es valida si se omite o es null, se usa la del documento que la creo. domain (opcional) String indicando el dominio para el cual es valida. Si se omite o es null, se usa el dominio del documento que la creo. secure (opcional) Valor booleano (true/false) que indica si la cookie necesita un canal seguro (HTTPS)*

Los primeros dos parámetros se requieren. Los otros, si se suministran, deben ser pasados en el orden listado mas arriba. Para omitir la opción no usada, se pasa null en su lugar. Por ejemplo, para llamar a SetCookie usando nombre, valor y ruta, usted tipea:

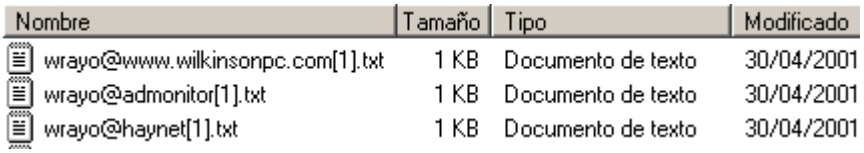
SetCookie ("myCookieName", "myCookieValue", null, "/");

Los parámetros omitidos al final no requieren pasar valores nulos. Para pasar una cookie segura para la ruta "/myPath", que expire después de la sesión actual:

SetCookie (myCookieVar, cookieValueVar, null, "/myPath", null, true);

DeleteCookie, *Funcion que borra una cookie (fija su fecha de expiracion a la fecha actual. name String que contiene el nombre de la cookie.*

Un cookie funciona, cuando visita por primera vez un sitio web que usa cookies, el servidor web (el computador que controla el sitio) envía al disco duro de su PC (a través del navegador) un cookie que contiene unos cuantos datos que le permitirán identificarlo. Cuando regresa al sitio, el navegador toma de su disco duro la cookie, y lo envía al servidor web. Así, él sabe quién es usted, y revisa en su base de datos la información que le permite personalizar ciertas opciones; si usted borra el cookie de su disco duro, el servidor web lo trataría como a un extraño porque no sabría que ha estado antes allí.



Nombre	Tamaño	Tipo	Modificado
wrayo@www.wilkinsonpc.com[1].txt	1 KB	Documento de texto	30/04/2001
wrayo@admonitor[1].txt	1 KB	Documento de texto	30/04/2001
wrayo@haynet[1].txt	1 KB	Documento de texto	30/04/2001

Existen tres grandes temores generados por las cookies: que saquen información de su PC, que los puedan leer desde cualquier sitio web o que puedan llevar virus.

- *Un cookie no puede leer información contenida en otro cookie ni en otro archivo del disco duro en el cual se encuentra almacenado.*
- *Cada cookie está marcado con información sobre el sitio que lo envió; un sitio web no puede leer información de cookies enviados por otros sitios.*
- *Según Netscape, "no es posible que un virus se pueda difundir a través de un cookie", ya que este no es más que un archivo de texto simple.*

3.3.3.3 Servidor Proxy

*Un **Proxy Server,** es un servidor destinado a almacenar páginas del Internet, gráficos, fotos y archivos que los usuarios usan mucho. Una vez que el usuario configuro el Proxy en su browser o navegador, toda conexión que intente realizar hacia Internet llegará primero al Proxy; si la información ya está allí almacenada, el mismo Proxy la entrega al usuario, de lo contrario, este contacta la fuente de los datos (en Internet), los pasa al cliente y los almacena para futuras consultas de otros usuarios.*

La característica más importante de un servidor Proxy es que dispone de una memoria caché muy grande, en la que guarda toda la información que le han solicitado a los usuarios, así cuando uno de ellos pide al servidor una dirección, primero verifica si la tiene en su registro, y si no consta en él, solicita la información al servidor de internet, por esto su principal ventaja es la reducción de tiempo.

Ventajas de un proxy

Las ventajas que ofrece la utilización de un proxy en una red local son las siguientes:

- *El programa y la instalación tienen un precio mucho menor que cualquier router.*
- *Sólo es preciso disponer de una línea telefónica normal o RDSI.*
- *La instalación emplea los dispositivos de la propia red local, por lo que se reduce a la configuración de los programas.*
- *El proxy también actúa como una barrera (firewall) adicional para limitar el acceso a la red local desde el exterior.*
- *El número IP es el que identifica a un ordenador en Internet. Si se utiliza un proxy basta un IP para toda la red local en lugar de un IP para cada uno de los ordenadores.*
- *No es necesario que el ordenador que actúa como proxy esté conectado permanentemente a Internet. Con esta función cada vez que un usuario realiza una petición el proxy establece la conexión.*

El proxy puede facilitar a los ordenadores de la red local la mayoría de los servicios de Internet, como:

- *· Correo electrónico*
- *· World Wide Web*
- *· Transmisiones FTP*
- *· Telnet*
- *· News*

Existen en el mercado servidores proxy que ofrecen además de estos servicios básicos otros servicios avanzados:

- *IRC*
- *Socks*
- *Real Audio*
- *Stream Works*
- *DNS*

4.3.4 Caso Práctico

Dentro del mercado existen algunos tipos de software que permiten configurar a un servidor como un Proxy, como por ejemplo:

***Wingate** es una solución software que permite conectar a cualquier usuario de su red local a internet mediante una única conexión telefónica (módem o RDSI). Wingate es básicamente un servidor de proxy múltiple, servidor de Telmet, FTP, SOCKS (v40) y otras funciones en un único paquete.*

***WinProxy 1.5** es una solución de software que permite conectar a 2 o más computadores PC que se encuentran en una Red a un acceso simultáneo a Internet a través de una sola cuenta de acceso, vía módem telefónico, conexión ISDN (RDSI), ADSL o TV Cable y donde cada equipo puede utilizar los distintos servicios Internet. De esta manera, en forma **simple** y **económica** varios PC de su empresa pueden compartir una misma conexión y una cuenta de acceso a Internet, obteniendo importantes ahorros en equipos, cuentas de acceso a Internet y servicio local medido (SLM).*

***Microsoft Proxy Server,** Internet Security and Acceleration (ISA) Server es el sucesor de Proxy Server 2.0. ISA Server va más allá de un servidor proxy al proporcionar un firewall empresarial y un servidor de alto rendimiento para memoria caché de Web y satisfacer los requerimientos de los entornos más exigentes de Internet.*

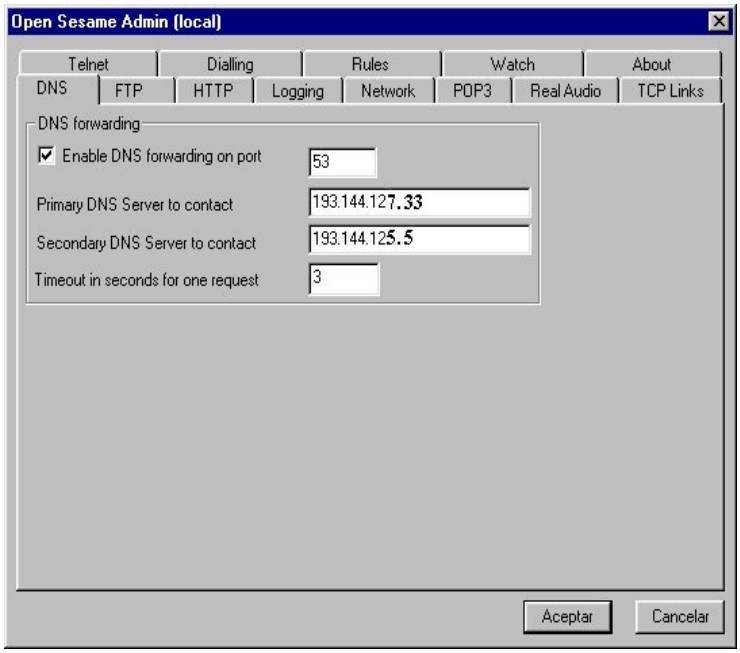
En este caso se seguirá la configuración del software OpenSesame para un server Proxy:

*En el menú, una vez instalado, dentro de Menú, Programas aparecerá un grupo denominado **Open Sesame**.*

El primer elemento del grupo es el programa que deberá estar en ejecución para que el servidor Proxy funcione, el segundo elemento es la herramienta de administración del Open Sesame, este programa es el que se debe ejecutar para poder configurar adecuadamente el servidor Proxy.

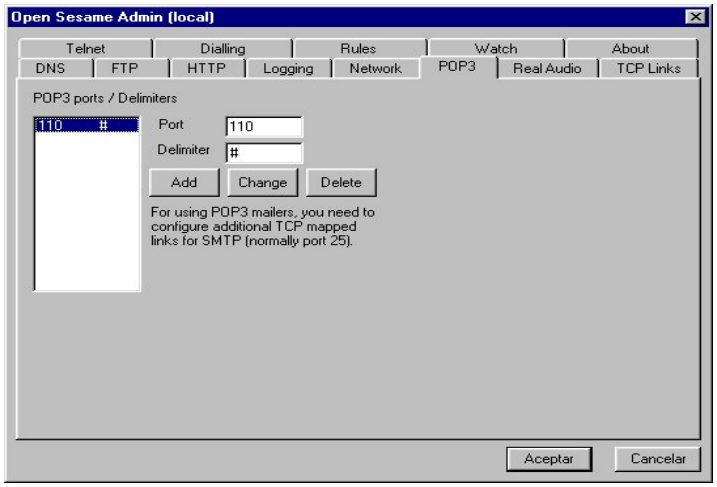
En la herramienta de administración se podrá visualizar una ventana con distintas opciones, una por cada uno de los elementos para configurar: DNS, FTP, HTTP, etc.

Para permitir la navegación *en Internet por medio del servidor Proxy únicamente será necesario modificar la configuración del elemento denominado* ***DNS****, dejándolo tal como se indica en la imagen.*

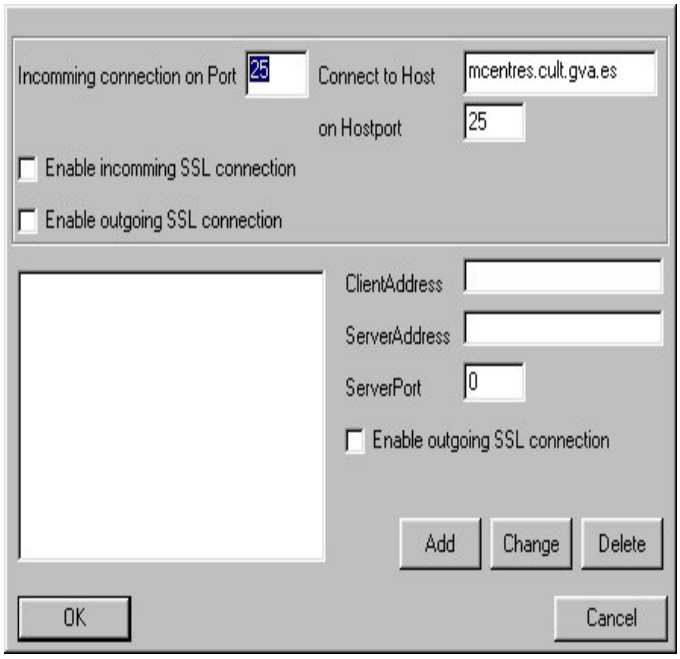


Para utilizar correo electrónico y *los programas cliente de correo electrónico desde los ordenadores no conectados diréctamente a Internet, será necesario configurar tanto el correo saliente (SMTP) como el correo entrante (POP3) en la herramienta de administración de Open Sesame anteriormente citada.*

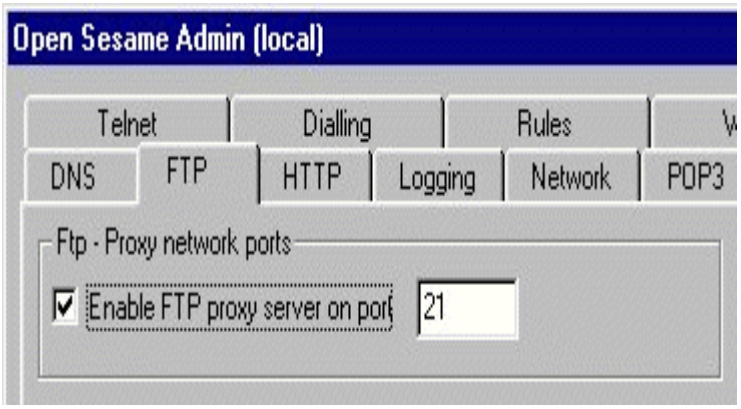
Correo Entrante, *hay que modificar el elemento denominado* ***POP3*** *tal y como se indica en la siguiente imagen, luego se especifica como número de puerto el 110 y como delimitador el símbolo #, se pulsa el botón* ***Add*** *para que la modificación tengan efecto añadiéndose a la lista de la izquierda.*



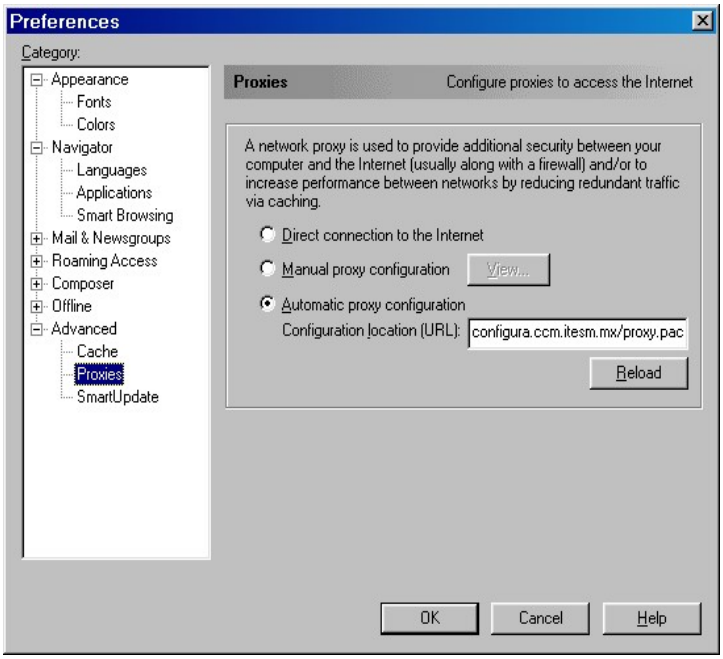
***Correo Saliente**, en el elemento **TCP links** (enlaces TCP) se da un clic sobre el botón **New** para establecer un nuevo enlace, con esto aparecerá una ventana en la que se deben especificar las configuraciones de puerto y servidor de correo que hay a continuación.*



***Para utilizar clientes FTP,** utilizar programas cliente de FTP desde los ordenadores no conectados directamente a Internet, debe colocarse en el elemento **FTP** de la herramienta de administración de Open Sesame y activar la casilla de verificación que aparece en dicha ventana.*



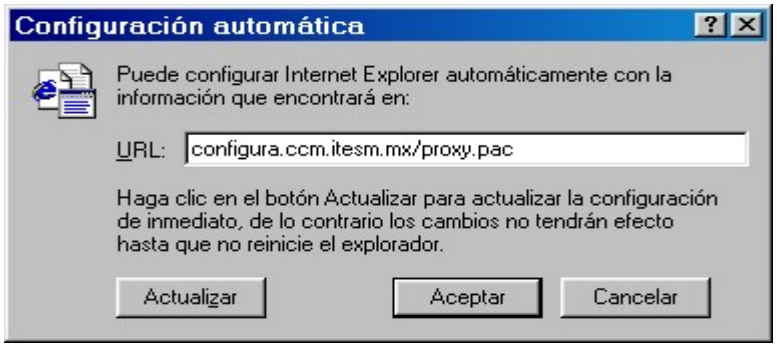
***Para las versiones recientes de Netscape,** En el menú Edición, Preferencias, Avanzadas, Proxy se debe habilitar Configuración automática del servidor Proxy.*



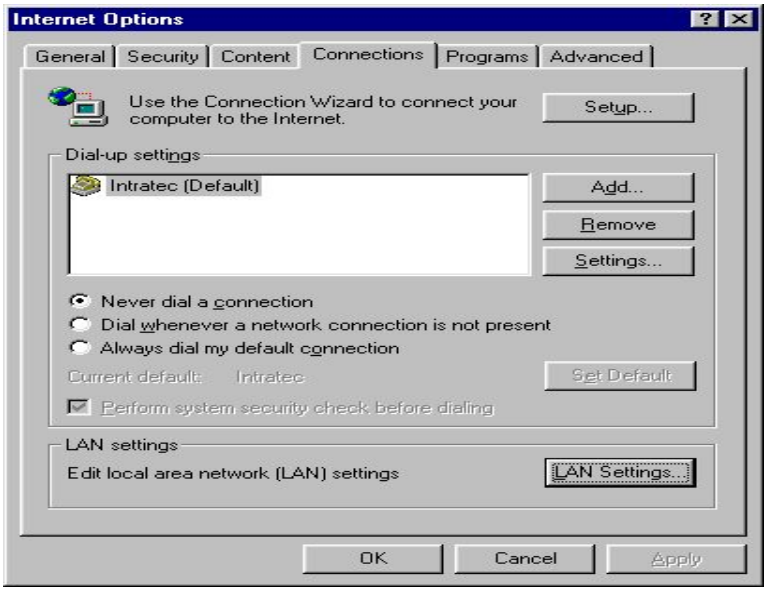
Hay un letrero que dice Dirección para configuración (URL): agrega la dirección URL que te correpnoda, en este caso es ***configura.ccm.itesm.mx/proxy.pac,*** *luego presiona el boton de Reload y finalmente el botón de Aceptar.*

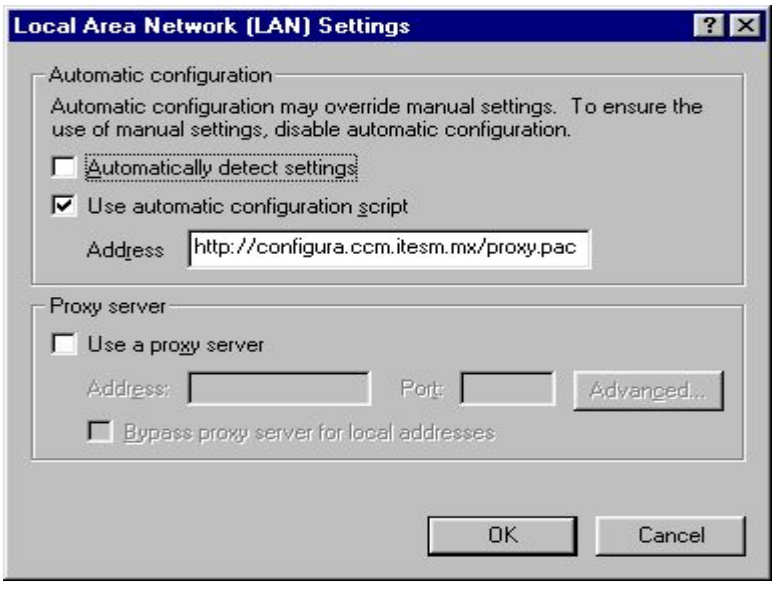
Para Microsoft Explorer 4*, En el menú Ver, Opciones, Conexión en la parte de Configuración Automática debe presionar el botón Configurar.*

Aparecerá una ventana donde tendrás que agregar en URL: ***configura.ccm.itesm.mx/proxy.pac*** *(la dirección que corresponda según el caso). Presione el botón de Actualizar y botón de Aceptar a las dos pantallas.*



Para Microsoft Explorer 5*, En el menú Herramientas, Opciones Internet,Conexiones, RED (LAN Settings), debe habilitar Usar script de configuración automática.*





La dirección en este caso es: http://configura.ccm.itesm.mx/proxy, presiona el botón de Aceptar a las dos pantallas y reinicie el navegador.

Este es un caso práctico para configurar un servidor proxy e ingrese a una institución X, es una forma de presentar los uno de los tantos programas que se encuentran en el mercado.

5. Conclusiones.

El protocolo IP es la base para el desarrollo de múltiples servicios que se están implementando en el mercado, pero que aun en nuestro medio son desconocidos o poco tratados. Este documento es una introducción al inmenso campo de lo que puede abarcar IP.
Vale la pena entonces profundizar en el tema y no esperar que la nueva tecnología, avances, información, o productos lleguen cuando sean obsoletos en el exterior.

Campos como la telefonía por IP proporcionan un método alternativo a la telefonía tradicional para la comunicación por voz, lo cual nos permitirá la integración de las redes de datos usadas para los ordenadores y la red telefónica para voz instaladas en los edificios, mediante lo que se consigue amortizar la primera al aprovechar el ancho de banda disponible que no se está utilizando, y además evita los costos de la instalación de la segunda; o como la videoconferencia por IP, que de igual forma nos brindan la facilidad y rapidéz de comunicarnos al otro lado del mundo, simplemente dos campos que son una mínima parte de lo que IP representará en un futuro muy cercano.

Bibliografía:

Andrew S. Tanenbaum, *Redes de Computadoras, Pearson Education Inc, 2003*

Douglas E. Comer*, Redes de Computadoras e Internet, 1996.*

Páginas Web

http://www.rediris.es/rediris/boletin/33/enfoque3.html
http://www.geocities.com/SiliconValley/Bay/8259/parte1.html#3
http://dmoz.org/World/Espa%F1ol/Computadoras/Internet/Protocolos/
http://www.openbsd.org.mx/~alex/openbsd_port_scanning/node11.html
http://ditec.um.es/laso/docs/tut-tcpip/3376c46.html#smtp
http://www.bellanet.org/email-s.htm
http://www.saulo.net/pub/articulo.php?cod=hosts
http://www.microsoft.com/latam/technet
http://club.telepolis.com/jlrosalesf/FUNDAMENTOS%20DEL%20TCP%20-22-.htm

Documentación otorgada por la Universidad Politécnica de Madrid.
Manuales de Windows NT 4.0
Revistas y Diccionarios Informáticos.

INDICE