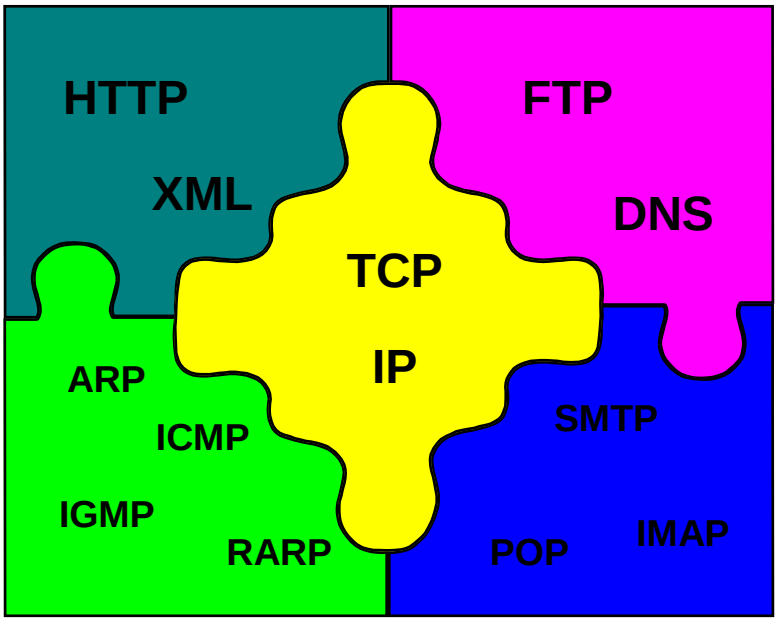
CAPITULO II

PROTOCOLOS TCP/IP



2.1. Introducción.

Aunque a TCP/IP lo confunden como si fuese un solo protocolo, en realidad es una colección de múltiples protocolos, de entre los cuales destacan el protocolo IP y el protocolo TCP. Se ha convertido en el protocolo más popular debido a que es utilizado por Internet y esta muy extendido en los sistemas operativos. Ver figura 2.1.

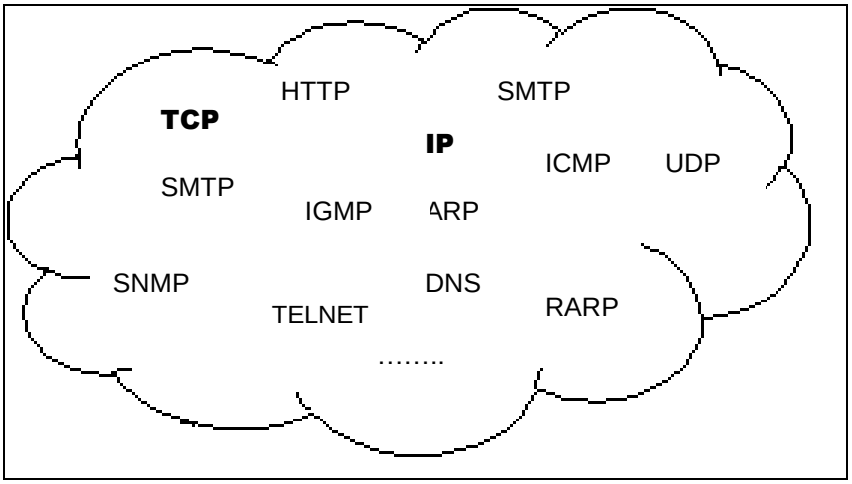


Figura 2.1: Conjunto de protocolos TCP/IP

TCP/IP se ha convertido en el conjunto de protocolos de red disponible más adaptable por el medio del cual se puede trabajar casi en cualquier medio de Red, Hardware y Sistema Operativo existente, desde una pequeña LAN de grupo de trabajo, hasta la conexión de millones de sistemas que componen la propia Internet.

¿De donde sale TCP/IP?

En 1969, la Agencia de proyectos de investigación avanzada sobre defensa (DARPA) subvencionó un experimento en el que se enlazaron tres computadoras. El objetivo de este proyecto era el de proporcionar una tecnología fiable de trabajo en red que pudiera recuperarse frente a problemas y errores. Originalmente se enlazaron tres sistemas entre sí con líneas alquiladas a la compañía telefónica y éstos utilizaban un protocolo llamado NCP (Protocolo de Control de Red, Network Control Protocol)

En 1973, la DARPA inició un programa de investigación de tecnologías de comunicación entre redes de diferentes características. El proyecto se basaba en la transmisión de paquetes de información, y tenia por objetivo la interconexión de redes. De este proyecto surgieron dos redes: Una de investigación, ARPANET, y una de uso exclusivamente militar, MILNET. Para comunicar las redes, se desarrollaron varios protocolos: El protocolo de Internet y los protocolos de control de transmisión. Posteriormente estos protocolos se englobaron en el conjunto de protocolos, los cuales dan lugar al modelo TCP/IP.

En 1980, se incluyo en el UNIX 4.2 de BERKELEY, y fue el protocolo militar estándar en 1983. Con el nacimiento en 1983 de INTERNET, este protocolo se populariza bastante, y su destino va unido al de Internet. ARPANET dejo de funcionar oficialmente en 1990.

Algunos de los motivos de su popularidad son:

- Independencia del fabricante
- Soporta múltiples tecnologías
- Puede funcionar en maquinas de cualquier tamaño
- Estándar de EEUU desde 1983
- Su destino está ligado a Internet

La arquitectura de un sistema en TCP/IP tiene una serie de metas:

- La independencia de la tecnología usada en la conexión a bajo nivel y la arquitectura del ordenador
- Conectividad Universal a través de la red
- Reconocimientos de extremo a extremo
- Protocolos estandarizados

2.2. Estructura Interna.-

El modelo ISO/OSI *(Organización de Estándares Internacionales para la Interconexión de Sistemas Abiertos)* utiliza capas para organizar una red dentro de módulos funcionales y bien definidos. Los diseñadores de redes utilizan las descripciones del modelo de estas capas para construir redes reales. En una red por capas, cada módulo (o capa) proporciona funcionalidad específica o servicios a sus capas adyacentes. Además cada capa esta protegida por otras capas arriba de ellas para detalles de implementación de nivel más bajo. Cada capa hace una interfaz sólo con la siguiente capa en la red.

La arquitectura de Internet esta basada en capas. Esto hace más fácil implementar nuevos protocolos. El conjunto de protocolos TCP/IP, al estar integrado plenamente en Internet, también dispone de este tipo de arquitectura. El modelo de capas de TCP/IP es algo diferente al propuesto por ISO/OSI (Organización de Estándares Internacionales para la Interconexión de Sistemas Abiertos OSI), ver Figura 2.2.

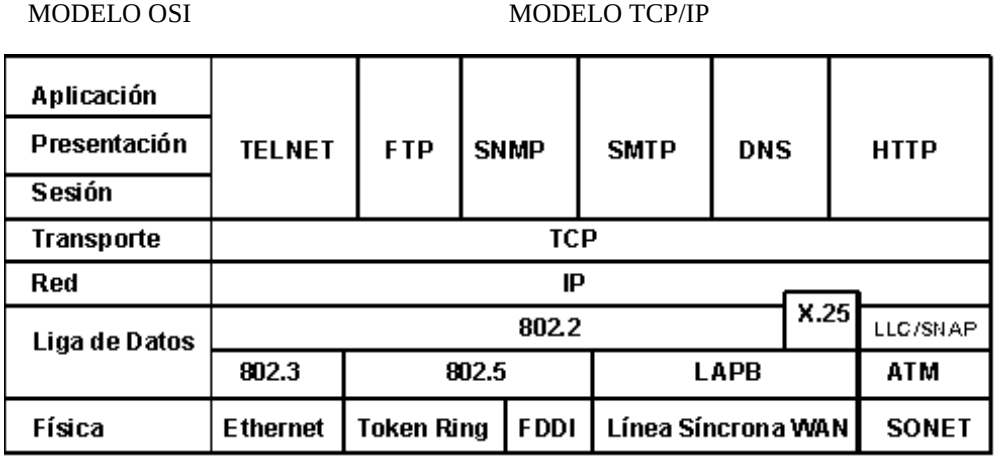


Figura 2.2: Relación del modelo TCP/IP con el modelo OSI

A continuación se analizará la función de cada capa en el modelo TCP/IP**.**

2.3 Capa Física

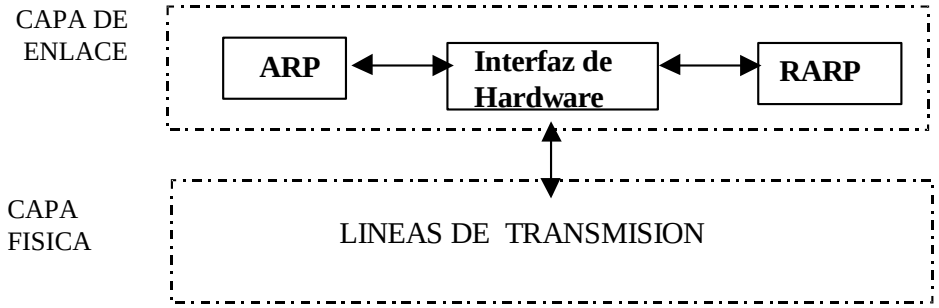


Figura 2.3: Capa f´ísica relacionada con la Capa de Enlace

Como se indica en la Figura 2.3 la capa física en una red TCP/IP es idéntica a la capa física del modelo ISO/OSI, la cual incluye el medio de transmisión que transporta los datos por la red. Este medio es casi siempre algún tipo de cable coaxial, par trenzado o fibra óptica. *El modelo TCP/IP no considera oficialmente el medio hardware como componente específico en su diseño*. TCP/IP tiende a agrupar la interfaz hardware con el nivel de interfaz de red.

Independientemente del medio hardware que se utilice, necesitará una tarjeta de interfaz de red específica. Estos dispositivos de interfaz de red son específicos del medio hardware por el que se transmiten las señales. Cada uno de estos servicios necesita un componente software llamado controlador de dispositivo. En la mayoría de los sistemas operativos de red, el controlador de dispositivo debe incluirse con el sistema operativo de base o proporcionarlo el fabricante del hardware.

2.4. Capa de Enlace.

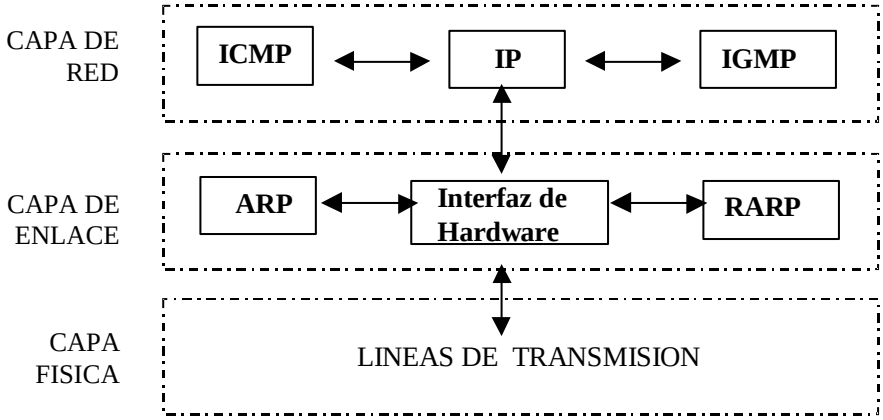


Figura 2.4: Capa de Enlace relacionada con Capa Física y Capa de Red

Como se indica en la Figura 2.4, la capa de enlace incluye una interfaz de Hardware y dos módulos de protocolos: El Protocolo de Resolución de Direcciones (ARP) y el Protocolo de Resolución de Direcciones Inverso (RARP).

Las direcciones Ethernet (a nivel físico) son de seis bytes de longitud, mientras las direcciones IP son de cuatro bytes. Todos los datos transmitidos a través de la red mediante tecnología Ethernet deben utilizar tramas de datos Ethernet; las tarjetas de interface Ethernet observan las tramas en la red en busca de sus propias direcciones Ethernet. Las tarjetas de interface no saben ni se preocupan por la dirección IP.

En otras palabras, los protocolos de TCP/IP sólo funcionan con direcciones IP; las tramas Ethernet con direcciones Ethernet. Estos diferentes tipos de direcciones representan un problema de comunicación en la red. Los protocolos de Resolución de Direcciones y de Resolución de Direcciones Inverso solucionan este problema analizando las direcciones: traducen las direcciones IP a direcciones de la capa de enlace y viceversa. Ver Figura 2.5.

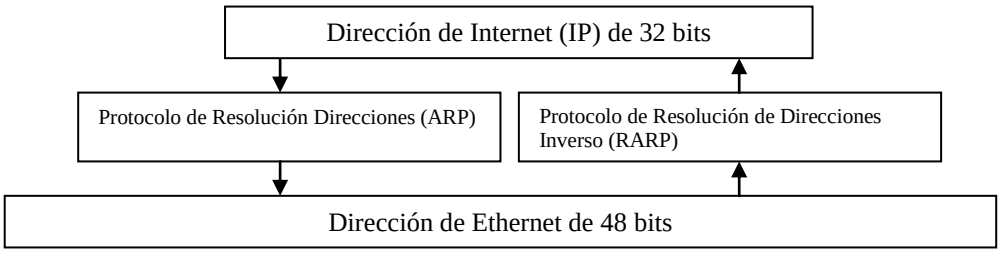


Figura 2.5: Traducción de direcciones IP a Ethernet

2.4.1. Protocolo resolución de direcciones (ARP)

El encaminamiento en el entorno de la red local utiliza el protocolo ARP que relaciona el nivel de red IP con los niveles inferiores. El protocolo ARP se usa para traducir las direcciones IP (lógicas) en direcciones de la red local (físicas).

El funcionamiento del protocolo ARP es muy simple. Cuando una máquina desea enviar un mensaje a otra conectada con ella a través de una red Ethernet, se encuentra con un problema: La dirección IP de la máquina en cuestión es diferente a la dirección física de la misma. La máquina que quiere enviar el mensaje sólo conoce la dirección IP del destino, por lo que tendrá que encontrar un modo de traducir la dirección IP a la dirección física. Esto se hace con el protocolo ARP.

Este protocolo utiliza una tabla denominada Tabla de Direcciones ARP, que contiene la correspondencia entre direcciones IP y direcciones físicas utilizadas recientemente. Si la dirección solicitada se encuentra en esta tabla el proceso se termina en este punto, puesto que la máquina que origina el mensaje ya dispone de la dirección física de la máquina destino.

Si la dirección buscada no esta en la tabla el protocolo ARP envía un mensaje a toda la red utilizando una dirección de “difusión”. Cuando un ordenador reconoce su dirección IP envía un mensaje de respuesta que contiene la dirección física. Cuando la máquina origen recibe este mensaje ya puede establecer la comunicación con la máquina destino, y esta dirección física se guarda en la Tabla de direcciones ARP.

La respuesta ARP contiene la dirección física y lógica del destinatario. La respuesta se enviará directamente al que origina la consulta ARP, que con esta información puede dirigir inmediatamente sus mensajes.

Las implementaciones del protocolo ARP incorporan Buffers con las tablas de correspondencia entre direcciones IP y direcciones físicas de la red, de forma que se reduce el número de consultas que se deben realizar.

2.4.2. Protocolo de Resolución de Direcciones Reversa (RARP)

El protocolo RARP (Reverse Address Resolution Protocol) es el encargado de asignar una dirección física a una dirección lógica (IP).

El formato del RARP es similar al del ARP. El valor del código de operación para una solicitud es 3, y el valor para una respuesta es 4.

Los desarrolladores de TCP/IP diseñaron RARP para que lo usaran computadoras sin disco duro. Por ejemplo, una estación de trabajo sin disco puede leer la dirección de su capa de enlace de su tarjeta de interfaz de red y solicitar a otro sistema que le cargue el sistema operativo.

2.5. Capa de Red

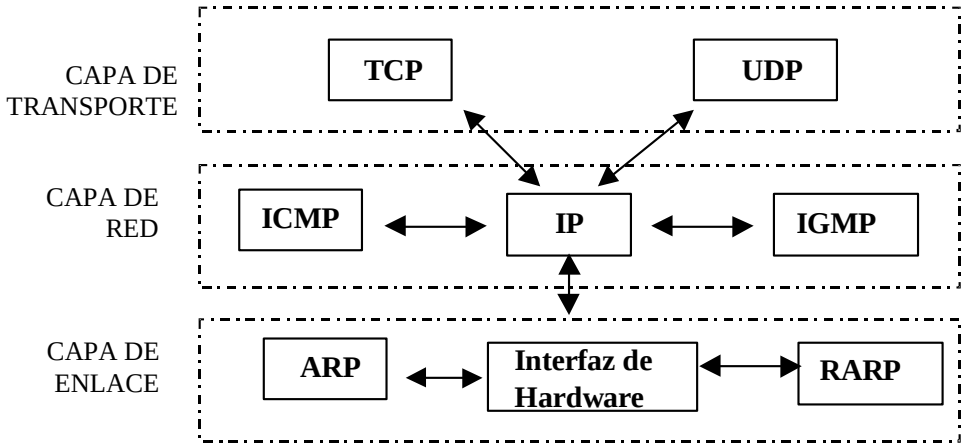


Figura 2.6 Capa de Red relacionada con la capa Enlace y capa de Transporte

La capa de red es el corazón de cualquier red basada en el protocolo TCP/IP. Esta capa incluye el protocolo Internet (IP), el protocolo de control

de mensajes de Internet (ICMP, Internet Control Message Protocol) y el protocolo de manejo de grupos de Internet (IGMP; Internet Group Management Protocol). IP hace casi todo el trabajo dentro de la capa de red. ICMP e IGMP son protocolos de apoyo para IP, pues lo ayudan a manejar los mensajes especiales de la red, como los de error y de transmisiones múltiples (mensajes enviados a dos o más sistemas), ver figura 2.6.

Además la capa de red controla la comunicación entre un equipo y otro. Conforma los paquetes IP que serán enviados por la capa inferior; Desencapsula los paquetes recibidos pasando a la capa superior la información dirigida a una aplicación.

2.5.1. IP (Internet Protocol)

En la actualidad se utiliza dos clases de direcciones IP la versión 4, y la versión 6. La versión 5 no paso de su fase experimental.

2.5.1.1. IP (Internet Protocol) versión 4

El Protocolo IP proporciona un sistema de distribución que es poco fiable incluso en una base sólida. El protocolo IP especifica que la unidad básica de transferencia de datos en el TCP/IP es el datagrama.

Los datagramas pueden ser retrasados, perdidos, duplicados, enviados en una secuencia incorrecta o fragmentados intencionadamente para permitir que un nodo con un buffer limitado pueda coger todo el datagrama. Es la responsabilidad del protocolo IP reensamblar los fragmentos del datagrama en el orden correcto. En algunas situaciones de error los datagramas son descartados sin mostrar ningún mensaje mientras que en otras situaciones los mensajes de error son recibidos por la máquina origen (esto lo hace el protocolo ICMP).

El protocolo IP también define cual será la ruta inicial por la que serán mandados los datos.

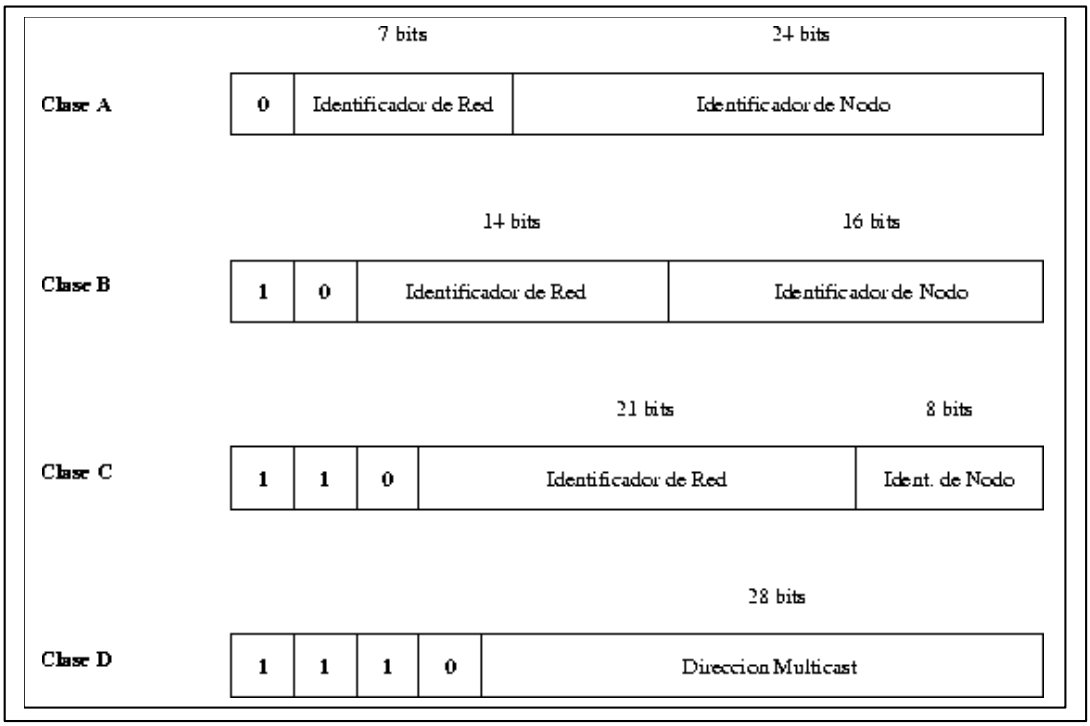
Cuando los datagramas viajan de unos equipos a otros, es posible que atraviesen diferentes tipos de redes. El tamaño máximo de estos paquetes de datos puede variar de una red a otra, dependiendo del medio físico que se emplee para su transmisión. A este tamaño máximo se le denomina MTU (Maximum Transmission Unit), y ninguna red puede transmitir un paquete de tamaño mayor a esta MTU. El datagrama consiste en una cabecera y datos.

Direcciones IP versión 4 (Ipv4)

Las direcciones IP hacen que el envío de datos entre ordenadores se haga de forma eficaz, de un modo similar al que se utilizan los números de teléfono.

Las direcciones IP tienen 32 bits, formados por cuatro campos de 8 bits separados por puntos. Cada campo puede tener un valor comprendido entre 0 y 255. Esta compuesta por una dirección de red, seguida de una dirección de subred y de una dirección de host.

Existen cinco clases de subredes, tal y como muestra la Figura 2.7 y se explica a continuación.



- La clase A contiene 7 bits para direcciones de red, con lo que permite tener hasta 128 redes, con 16.777.216 ordenadores cada una. Las direcciones estarán comprendidas entre 0.0.0.0. y 127.255.255.255. y la mascara de subred será 255.0.0.0.

- La clase B contiene 14 bits para direcciones de red y 16 bits para direcciones de hosts. El número máximo de redes es 16.536 redes, con 65.536 ordenadores por red. Las direcciones estarán comprendidas entre 128.0.0.0. y 191.255.255.255., y la mascara de subred será 255.255.0.0.

- La clase C contiene 21 bits para direcciones de red y 8 para hosts, lo que permite tener un total de 2.097.142 redes, cada una de ellas con 256 ordenadores. Las direcciones estarán comprendidas entre 192.0.0.0. y 223.255.255.255. y la mascara de subred será 255.255.255.0.

- La clase D se reserva todas las direcciones para multidestino (multicast), es decir, un ordenador transmite un mensaje a un grupo especifico de ordenadores de esta clase. Las direcciones estarán comprendidas entre 224.0.0.0. y 239.255.255.255.

- La clase E se utiliza exclusivamente para fines experimentales. Las direcciones están comprendidas entre 240.0.0.0. y 247.255.255.255.

Además, el software de Internet interpreta un campo con todos los bits en 1 como “all” (todos). Un campo de dirección que contiene todos 1 representa una dirección de emisión (o, en otras palabras, un mensaje destinado para todas las computadoras en la red). Normalmente, el software de Internet interpreta un campo con todos 0 (all 0`s) como “this” (esta). En otras palabras, un campo de dirección con todos 0 representará “esta” red y “esta” computadora anfitrión. Internet reserva estas dos direcciones (all 1´s y all 0´s) sólo para estos propósitos.

2.5.1.2. IP (Internet Protocol) Versión 6

Esta es una nueva versión del protocolo IP, llamada IPv6, aunque también es conocida como IPng (Internet Protocol Next Generation). Es la versión 6, debido a que la número 5 no pas de la fase experimental. La compatibilidad con la versión 4 es prácticamente total, ya que se han incluido características de compatibilidad. Algunas de las modificaciones, están encaminadas a mejorar la seguridad en la red, que apenas existía en la versión 4.

Direcciones IP Versión 6 (Ipv6)

El cambio más significativo en las direcciones ha sido, que ahora, se refieren a una interfaz y no a un nodo, aunque como cada interfaz pertenece a un nodo, es posible referirse a estos mediante su interfaz.

El número de direcciones diferentes se ha multiplicado de una manera exagerada. Teóricamente, es posible tener 2^{128} direcciones diferentes. Este número quiere decir que se podrían llegar a tener mas de 665.000 trillones de direcciones por metro cuadrado, aunque si siguieran una jerarquía, este número decrece hasta 1564 direcciones por metro cuadrado en el peor caso o tres trillones siendo optimistas.

En el IPv6 existen tres tipos básicos de direcciones:

- Direcciones unicast: Están dirigidas a una única interfaz en la red. Actualmente se dividen en varios grupos, y existe un grupo especial que facilita la compatibilidad con las direcciones de la versión 4.
- Direcciones anycast: Identifican a un conjunto de interfaces de red. El paquete se enviara a cualquier interfaz que forme parte del conjunto. En realidad son direcciones unicast que se encuentran asignadas a varias interfaces.

- Direcciones multicast: Identifican a un conjunto de interfaces de la red, de manera que cada paquete es enviado a cada uno de ellos individualmente.

2.5.2. Fragmentación.

Las tecnologías de red, tales como Ethernet, especifican una Unidad de transferencia Máxima (MTU). La MTU define el tamaño máximo del paquete que puede transmitir la red. Cuando una aplicación transmite el paquete más grande que la MTU de la red, el software de red automáticamente divide el paquete en pedazos más pequeños y transmite los datos como múltiples paquetes. Los campos candidatos a ser fragmentados del encabezado IP, tales como identificación, Banderas y Reproducción de fragmentos, son actualizados para indicar que el paquete es un fragmento y en que orden debe ser reensamblado.

Cuando el anfitrión destino recibe los paquetes IP fragmentados, un contador de reensamble se inicia. Todos los fragmentos deben llegar antes que el contador expire, de otra forma el anfitrión descartará todos los fragmentos. Debido a que la fragmentación y reensamble ocurren entre las capas de red y enlace de su red, el proceso es normalmente transparente.

2.5.3. Internet Control Message Protocol (ICMP)

Internet es un sistema autónomo que no dispone de ningún control central. El protocolo ICMP (Internet Control Message Protocol), proporciona el medio para que el software de hosts y gateways intermedios se comuniquen. El protocolo ICMP tiene su propio número de protocolo (número 1), que lo habilita para utilizar el IP directamente. La implementación de ICMP es obligatoria como un subconjunto lógico del protocolo IP. Los mensajes de error de este protocolo los genera y procesa TCP/IP, y no el usuario

ICMP provee reporte de mensajes y errores. Por ejemplo, si IP no es capaz de entregar un paquete en el host destino, ICMP envía un mensaje de "destino no encontrado" (destination unreachable) al nodo emisor.

Los mensajes más comunes de ICMP son:

Mensaje	Tipo	Función
Echo request	8	Usado por PING para encontrar un host.
Echo reply	0	Usado por PING para confirmar que un nodo ha sido encontrado.
Redirect	5	Informa al nodo de una ruta preferida.
Source quench	4	Informa al nodo disminuir la cantidad de datagramas debido a congestionamiento en la red.
Destination unreachable	3	Informa al nodo que el datagrama no pudo ser entregado.

Los mensajes de ICMP están contenidos en datagramas IP. Esto asegura que el mensaje ICMP será ruteado al nodo apropiado. El destino de un mensaje ICMP es siempre un módulo de software de la capa de red. El módulo ICMP en la capa IP del destino determina si debe pasar el mensaje a cualquiera de los módulos de software del nivel superior.

ICMP sólo proporciona servicios para notificar errores, es decir que no proporciona ningún servicio de corrección de errores, además, no especifica ninguna acción que los módulos de software de la capa de red deben tomar en respuesta a los errores que reporta.

2.5.4. Protocolo de Manejo de Grupos De Internet (IGMP)

EL IGMP (Internet Group Management Protocol) es un protocolo que funciona como una extensión del protocolo IP. Se utiliza exclusivamente por los miembros de una red multicast para mantener su status de miembros, o para propagar información de direccionamiento.

Un Gateways multicast manda mensajes una vez por minuto como máximo. Un Host receptor responde con un mensaje IGMP, que marca al Host como

miembro activo. Un Host que no responde al mensaje se marca como inactivo en las tablas de direccionamiento de la red multicast.

Para aplicaciones como conferencias interactivas, se utiliza la transmisión múltiple, y de esta manera se puede enviar información a varios pero no necesariamente a todos los receptores en la red. Los anfitriones y los receptores que soportan la transmisión múltiple usan el módulo del Protocolo de Manejo de Grupos De Internet (IGMP)

2.6. Capa de Transporte.

Para comunicarse con Internet, las aplicaciones que se utilizan intercambian datos con la capa de transporte TCP/IP. Esta incluye dos protocolos de transporte: El Protocolo de Control de Transporte (TCP*, Transport Control Protocol*) y el Protocolo de Datagrama de Usuario (UDP*, User Datagrama Protocol).* Ver figura 2.8

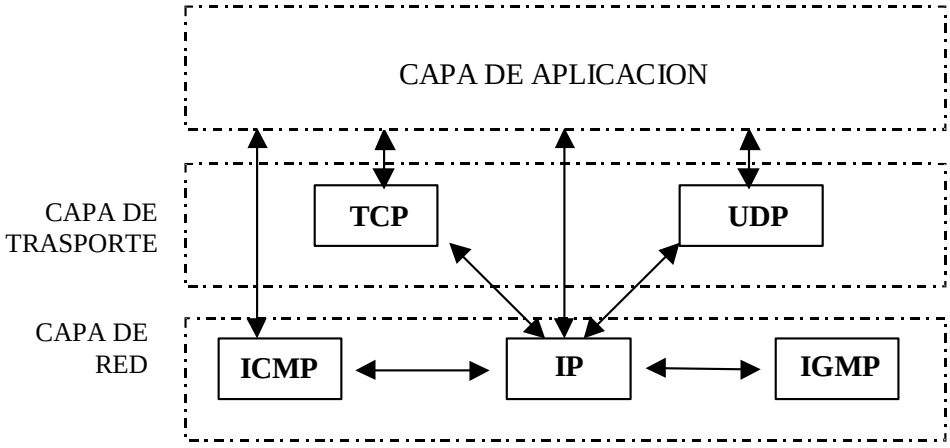


Figura 2.8**: Capa de transporte relacionada con las Capa de Red y Capa de Aplicación**

El protocolo de control de transmisión (TCP) es *un protocolo orientado a conexión* que utiliza un flujo de bytes confiable para enviar y recibir datos; este protocolo proporciona un circuito virtual para comunicaciones de red. El protocolo de Datagrama de Usuario es un protocolo no confiable, sin conexión que utiliza datagramas para enviar y recibir datos.

Un puerto es como una dirección IP, excepto que TCP/IP asocia un puerto a un protocolo en lugar de a una computadora anfitrión. En la misma forma que los datagramas IP almacenan direcciones IP fuente y destino, los protocolos de transporte almacenan números de puerto fuente y destino. En pocas palabras, los programas de red asocian un puerto de protocolo Internet con una aplicación y función específicas.

Como protocolo sin conexión y no confiable, UDP simplemente deposita datos en el puerto. UDP no mantiene una conexión entre el emisor y el receptor. En contraste, TCP esta orientado a conexión. TCP mantiene una conexión mientras sé esta comunicando. Además, TCP puede abrir múltiples conexiones en el mismo puerto.

2.6.1. Protocolo de Datagrama de Usuario (UDP)

El protocolo UDP (User Datagram Protocol) proporciona aplicaciones con un tipo de sevicio de datagramas orientado a transacciones. El servicio es muy parecido al protocolo IP en el sentido de que no es fiable y no esta orientado a la conexión. El UDP es simple, eficiente e ideal para aplicaciones como el TFTP y el DNS. Una dirección IP sirve para dirigir el datagrama hacia una maquina en particular, y el número de puerto de destino en la cabecera UDP se utiliza para dirigir el datagrama UDP a un proceso específico localizado en la cabecera IP. La cabecera UDP tambien contiene un número de puerto origen que permite al proceso recibido conocer como responder al datagrama.

2.6.2. Protocolo de Control de Transmisión TCP.

El protocolo TCP con el protocolo IP son los que con mayor frecuencia se utilizan en el conjunto de protocolos TCP/IP (de ahí el nombre)

Al igual que el Protocolo de Datagrama de Usuario(UDP), TCP transporta datos entre las capas de red y de aplicación, pero es mucho más complejo que UDP, pues proporciona un servicio de entrega de datos confiable,de

flujo de bytes y orientado a conexión; en otras palabras, TCP asegura la entrega, así como también se encarga que la aplicación destino reciba los datos en la secuencia correcta. En contraste, UDP no garantiza la entrega de datagramas, ni que estos lleguen en la secuencia adecuada.

TCP también intenta optimizar el ancho de banda de la red. Para hacerlo, controla dinámicamente el flujo de datos entre las conexiones. Por lo tanto, si el buffer de datos en el lado receptor de la conexión TCP comienza a sobrecargarse, TCP indica al lado emisor que reduzca la velocidad de transmisión.

2.6.2.1. Interfaces TCP.

Existen dos tipos de interfaces entre la conexión TCP y los otros programas. El primero es utilizar la pila de los programas de la capa de red. Como en esta capa solo esta el protocolo IP, la interfaz lo determina este protocolo. El segundo tipo es el interfaz del programa de usuario. Esta interfaz puede variar según el sistema operativo, pero en general tiene las siguientes características.

La interfaz envuelve el programa de usuario llamando a una rutina que introduce entradas en una estructura de datos llamada el Bloque de Control de Transmisión (TCB). Las entradas se realizan inicialmente en la pila de hardware y transferidas al TCB por medio de una rutina de sistema. Estas entradas permiten al TCP asociar un usuario con una conexión particular, de modo que pueda aceptar comandos de un usuario y mandarlos a otro usuario en la otra parte de la conexión. TCP utiliza unos identificadores únicos para cada parte de la conexión. Esto se utiliza para recordar la asociación entre dos usuarios. Al usuario se le asigna un nombre de conexión para utilizarlo en futuras entradas del TCB. Los identificadores para cada extremo de la conexión se llaman sockets. El socket local se construye concatenando la dirección IP de origen y el número de puerto de origen. El socket remoto se obtiene concatenando la dirección IP de destino y él número de puerto de destino.

El par de sockets de una conexión forman un único número en Internet. El UDP tiene los mismos sockets, pero no los recuerda. Esta es la diferencia entre un protocolo orientado a conexión y otro a no conexión. A continuación se explican los comandos más usuales:

- ***Open*:** Inicia una conexión o comienza a escuchar un socket. El usuario tiene un nombre de conexión local que actúa como un puntero dentro del TCB.
- ***Send*:** El comando Send manda datos del buffer especificado.
- ***Receive:*** El comando Receive es un mensaje de error si el nombre local proporcionado no es utilizado antes con el comando Open.
- ***Close:*** El comando Close hace que se cierre una conexión. Se produce un error si la conexión especificada no ha sido abierta, o si no se tiene autorización para cerrar la conexión.
- ***Status***: El comando Status solo tiene una variable asociada, que es el nombre de la conexión.
- ***Abort:*** El comando Abort hace que todos los comandos Send y Receive asociados al nombre de la conexión local se interrumpan. La entrada del usuario del TCB se elimina y se envía un mensaje especial de reinicio a la entidad del otro lado de la conexión.

El TCP recuerda el estado de cada conexión por medio del TCB. Cuando se abre una conexión, se efectúa una entrada única en el TCB. Un nombre de conexión se le asigna al usuario para activar los comandos de la conexión. Cuando se cierra una conexión se elimina su entrada del TCB.

2.6.2.2. Control de Flujo

El protocolo TCP puede controlar la cantidad de datos que debe enviar mediante el campo Window. Este campo indica el número máximo de octetos que pueden ser recibidos. El receptor de un segmento con el campo window a cero, no puede enviar mensajes al emisor, excepto mensajes de prueba. Un mensaje de prueba es un mensaje de un solo octeto que se utiliza para detectar redes o hosts inalcanzables.

2.6.2.3. Formato del segmento TCP

La figura 2.9 muestra la estructura del encabezado TCP.

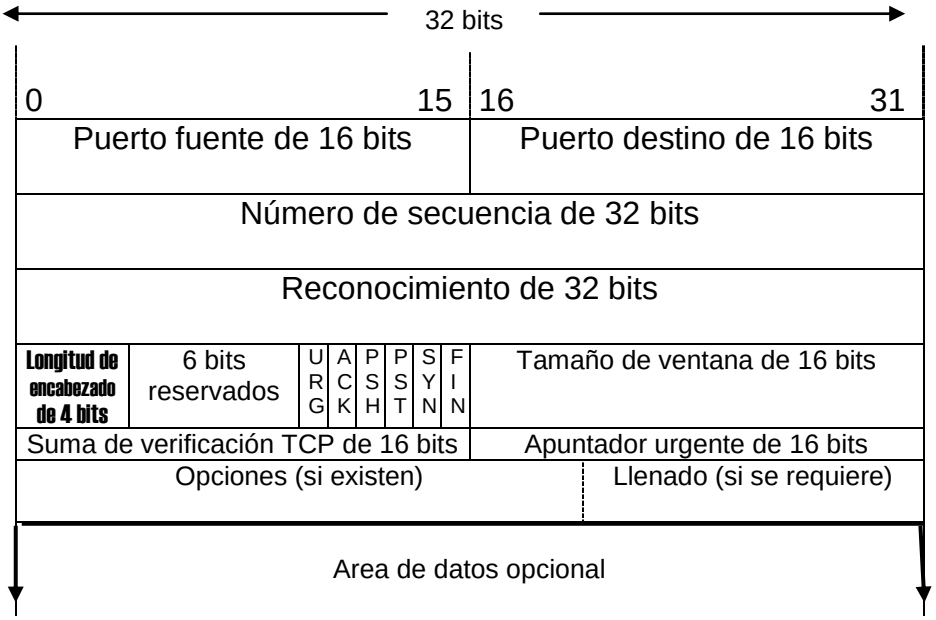


Figura 2.9: Estructura del encabezado TCP

- **Puerto Fuente.**- Identifica el puerto del protocolo de la aplicación emisora.
- **Puerto Destino.-** Identifica el puerto del protocolo de la aplicación receptora.
- **Número de Secuencia.-** Identifica el byte de primer orden de datos en el área de datos del segmento TCP.
- **Número de Reconocimiento.-** Identifica el próximo byte de datos que el emisor espera del flujo de datos.
- **Longitud de encabezado.**- Especifica la longitud del encabezado TCP en palabras de 32 bits
- **Bandera URG.-** Indica el módulo TCP receptor que el campo apuntador urgente apunte hacia datos urgentes
- **Bandera ACK.-** Indica el módulo TCP receptor que el campo Número de reconocimiento contiene el número de reconocimiento válido.
- **Bandera PSH.-** Indica al módulo TCP receptor que envíe inmediatamente los datos a la aplicación destino.
- **Bandera RST.-** Pide al módulo TCP receptor que restablezca la conexión TCP
- **Bandera SYN.-** Indica al módulo TCP receptor que sincronice los números de secuencia.
- **Bandera FIN.-** Indica al módulo TCP receptor que el emisor ha terminado de enviar los datos**.**
- **Tamaño de ventana.-** Indica al módulo TCP receptor el número de bytes que el emisor está dispuesto a aceptar.
- **Suma de verificación TCP.-** Ayuda al módulo TCP receptor a detectar corrupción de datos.
- **Apuntador urgente.-** Apunta al último byte de datos urgentes en el área de datos TCP.
- **Opciones.-** Usualmente utilizado con la opción Tamaño máximo de segmento, la cual informa el segmento más grande que el módulo TCP espera recibir.

2.6.2.4 Estados del TCP

El inicio, mantenimiento y cierre de una conexión requiere que el TCP recuerde toda la información relativa a cada conexión. Esta información se almacena en una entrada para cada conexión dentro del TCB. Cuando se abre una conexión, la entrada en el TCB se realiza con todas las variables inicializadas con sus respectivos valores. Durante la conexión, la entrada del TCB es actualizada a medida que cambia la información. A continuación se describen algunos de los estados del TCP:

0. **CLOSED:** No existe, solo para referencia.
1. **LISTEN**: Esperando solicitud de conexión de un TCP remoto.
2. **SYN-SEN:** Esperando un mensaje de solicitud de conexión después de haber enviado una solicitud de conexión.
3. **SYN-RECEIVED:** Esperando confirmación de una reconocimiento de solicitud de conexión, después de haber enviado y recibido una solicitud de conexión.
4. **ESTABLISHED**: Representa una conexión abierta. Los datos recibidos pueden ser enviados a un protocolo de una capa superior. Este es el estado normal de la fase de transferencia de la conexión.
5. **FIN-WAIT-1:** Esperando la solicitud de fin de conexión de un TCP remoto, o un reconocimiento de una solicitud de fin de transmisión enviada anteriormente.
6. **FIN-WAIT-2:** Esperando una solicitud de fin de conexión de un TCP remoto.
7. **CLOSE-WAIT:** Esperando una solicitud de fin de conexión de un protocolo de una capa superior.
8. **CLOSING:** Esperando el conocimiento de una solicitud de final de conexión de un TCP remoto.
9. **LAST-ACK**: Esperando el conocimiento de una solicitud de final de conexión enviada anteriormente al TCP remoto.
10. **TIME-WAIT:** Esperando el tiempo necesario para que el TCP remoto haya recibido el conocimiento de la solicitud del fin de conexión.

2.7. Capa de Aplicación.

Este es el punto en que el modelo OSI y el modelo TCP/IP empiezan a ir por caminos separados. TCP/IP reconoce cualquier cosa a partir de este punto como un protocolo de aplicación, mientras que el modelo OSI descompone aún más las descripciones. Ver figura 2.10

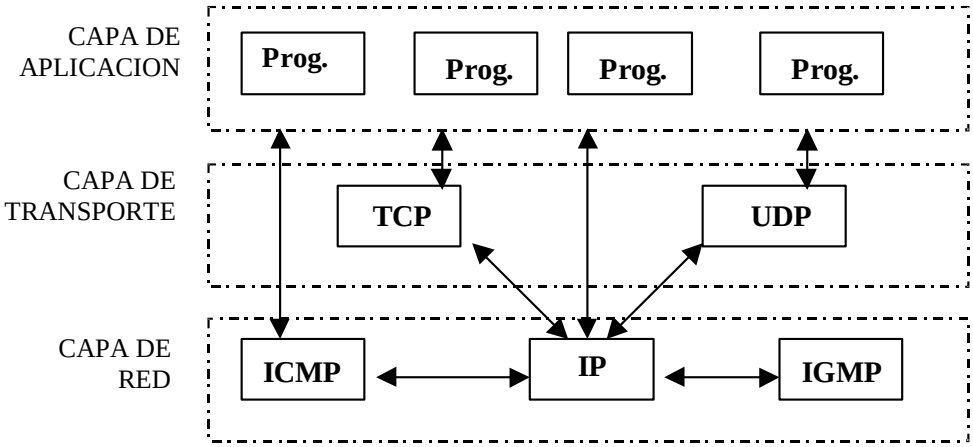


Figura 2.10: Relación entre la Capa de Aplicación con capas inferiores

En la actualidad se utilizan cientos de protocolos de aplicación. Algunos están todavía en fase experimental, otros están en espera de un reconocimiento formal. A continuación se tratan algunos de los protocolos más comunes y su utilización.

2.7.1. TELNET.

Las aplicaciones de acceso remoto, también llamadas TELNET (Telecommunicating Networks) nos permiten acceder a un servidor emulando un terminal que se encontrase físicamente conectado a él.

Manteniendo la arquitectura cliente - servidor presente en Internet, se necesita disponer de un programa cliente de TELNET que será el encargado de entenderse con el programa servidor de TELNET que estará ejecutando en la máquina remota. Normalmente se trata de máquinas con sistema operativo UNIX y como se trata de programas o procesos que se ejecutan de

forma desatendida, se les suele denominar demonios (daemons, en inglés). El programa cliente de TELNET es un programa que ofrece un entorno no gráfico, es decir, modo carácter, ya que aunque se utilice dentro de un entorno de ventanas, la funcionalidad se consigue mediante el uso de comandos del sistema operativo.

Normalmente, el programa servidor al que se quiere acceder está escuchando en el puerto 23 por defecto. No siendo necesario indicárselo al cliente TELNET que vamos a usar. De no ser así, es necesario indicar el puerto en el que está el proceso servidor, además de la dirección Internet del Servidor.

Puesto que el efecto del servicio de acceso remoto es presentarnos en local la máquina remota, será necesario disponer de cuenta como usuario de la máquina a la que se quiere acceder y además será muy conveniente tener unos conocimientos mínimos del sistema operativo que utilice la máquina remota, ya que serán estos los comandos que necesitemos utilizar para realizar cualquier tarea.

2.7.2. FTP. (File Transfer Protocol o Protocolo para Transmisión de Archivos)

Un servidor FTP funciona como un gran disco duro con directorios o divisiones en el cual si se tiene permiso asignado se puede enviar archivos o copiar los que allí existen.

Al usar un programa de FTP, se elige primero el computador que se desea acceder y se identifica con un nombre y contraseña. Una vez conectados, se presenta una lista de los archivos y directorios disponibles en dicho sitio.

Numerosos sitios FTP mantienen una parte abierta al público que se puede acceder usando el nombre "**anonymous**" como usuarios anónimos y la dirección de correo-electrónico como contraseña.

Los servidores FTP son una forma conveniente de hacer disponibles al público informaciones, resúmenes de discusiones, investigaciones, programas y actualizaciones de software. Por medio de los sitios FTP los fabricantes de software y hardware distribuyen documentos sobre los problemas confrontados por otros usuarios y cómo solucionarlos, incluyendo "parches" y actualizaciones para sus productos.

2.7.3. HTTP (HyperText Transfer Protocol o Protocolo para la Transferencia de HiperTexto)

HTTP es el conjunto de reglas para la transmisión y recepción de documentos hipertexto. Es usado por la WWW desde 1990 y es el protocolo responsable del entendimiento universal de las páginas de la WWW escritas en HTML. Fue ideado por Tim Berners-Lee (padre de la WWW) y sus especificaciones técnicas escritas por él, en conjunto con Roy T. Fielding, y Henrik Frystyk Nielsen, están disponibles al público en la red. (RFC 1945)

Una de las principales debilidades del HTTP (y de Internet en general) es que carece de facilidades de seguridad para la información transmitida y como respuesta a esto, ha surgido varias soluciones particulares, en especial el HTTP Seguro (Secure HTTP en inglés o S-HTTP), el cual goza de bastante popularidad.

Los recursos u objetos que actúan como entrada o salida de un comando HTTP están clasificados por su descripción MIME. De esta forma, el protocolo puede intercambiar cualquier tipo de dato, sin preocuparse de su contenido. La transferencia se realiza en modo binario, byte a byte, y la identificación MIME permitirá que el receptor trate adecuadamente los datos.

Las principales características del protocolo HTTP son:

Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original. Permite la

transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.

Existen tres verbos básicos (hay más, pero por lo general no se utilizan) que un cliente puede utilizar para dialogar con el servidor:

- **GET**, para recoger un objeto,
- **POST**, para enviar información al servidor y
- **HEAD**, para solicitar las características de un objeto (por ejemplo, la fecha de modificación de un documento HTML).

Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto. No mantiene estado. Cada petición de un cliente a un servidor no es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto.

Cada objeto al que se aplican los verbos del protocolo está identificado a través de la información de situación del final de la URL.

HTTP se diseñó específicamente para el World Wide Web: es un protocolo rápido y sencillo que permite la transferencia de múltiples tipos de información de forma eficiente y rápida. Se puede comparar, por ejemplo, con FTP, que es también un protocolo de transferencia de ficheros, pero tiene un conjunto muy amplio de comandos, y no se integra demasiado bien en las transferencias multimedia.

Etapas de una transacción HTTP

Para profundizar más en el funcionamiento de HTTP, veremos primero un caso particular de una transacción HTTP. Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo "Location " del cliente Web. El cliente Web descodifica la URL, separando sus

diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.

- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente. Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD,...), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor,...

- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información. Se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP. Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.

En la actualidad se ha mejorado este procedimiento, permitiendo que una misma conexión se mantenga activa durante un cierto periodo de tiempo, de forma que sea utilizada en sucesivas transacciones. Este mecanismo, denominado HTTP Keep Alive, es empleado por la mayoría de los clientes y servidores modernos. Esta mejora es imprescindible en una Internet saturada, en la que el establecimiento de cada nueva conexión es un proceso lento y costoso.

2.7.4. NNTP. (Network News Transfer Protocol o Protocolo de Transferencia de Noticias por Red)

NNTP (Network News Transfer Protocol) es un protocolo para la distribución, petición, recuperación y envío de news (noticias) entre los servidores de news de la comunidad ARPA-Internet a través de USENET utilizando un

sistema cliente-servidor con intercambio fiable de información (TCP). Está diseñado de forma y manera que los artículos están en un host, que actúa de servidor de news, y los usuarios "enganchados" a otros hosts de la misma red pueden acceder a los artículos a través de la conexión de éstos con el servidor central de news.

Normalmente un servidor NNTP corre como un proceso en background en un host, y acepta, si es posible, las conexiones de otros hosts de la red LAN en la que se encuentre. Esto funciona bien cuando hay un número de sistemas de ordenadores pequeño (con pocos usuarios), y un servidor central grande. De esta manera NNTP permite una manera bastante eficiente de acceso a las news basada en la cooperación entre los hosts de una misma red LAN u otras redes rápidas.

Utilizando NNTP, el intercambio de news entre hosts se hace de manera interactiva para poder indicar que artículos serán transmitidos. Si uno de los hosts quiere conseguir nuevas news, o si las quiere enviar él, contacta con uno o más de sus vecinos de la red. Para ello primero pregunta si ha sido creado algún nuevo grupo en el servidor central utilizando en comando NEWGROUPS.

Después el cliente preguntará por los nuevos artículos que hayan podido llegar a los grupos ya existentes de los que desea recibir news. Esto lo hace con el comando NEWNEWS y recibirá como respuesta una lista de los nuevos artículos desde el servidor, y así el cliente puede pedir la transmisión de aquellos artículos que desea y que no tiene ya.

Finalmente, el cliente puede avisar al servidor de las nuevas news que le hayan podido llegar de los usuarios que a él estén enganchados. El servidor le indicará si alguno de ellos ya le ha llegado por otro host y así el cliente solo le mandará los que no tenga ya. De esta manera se puede observar que el tráfico de información se reduce al mínimo necesario, con lo que aumenta la eficiencia.

De esta manera aquellos sitios que cuentan con un gran número de estaciones de trabajo encuentran este sistema muy conveniente para permitir a sus usuarios utilizar el servicio de news sin tener que almacenar el software y los artículos en cada estación. Además dada la garantía de rápida propagación de las news (muy, muy rápida) que NNTP ofrece hace que sea un sistema muy utilizado en la mayoría de servidores USENET.

NNTP permite:

- A un servidor, obtener noticias de otro servidor de noticias
- A un agente de noticias de cliente, obtener noticias de un servidor de noticias
- A un agente de noticias del cliente, enviar un nuevo artículo al servidor de noticias.

Para acceder a los artículos de noticias, el proceso cliente se conecta al puerto 119 del servidor de noticias. El cliente envía una serie de comandos y recibe las respuestas.

2.7.5. SMTP (Simple Mail Trasport Protocol o Protocolo Simple de Transferencia de Correo)

Dirección de Correo

La dirección de correo tiene la forma de una cuenta (un espacio en un servidor) y un nombre de dominio, separados por el caracter especial @, el nombre de dominio está especificado en el URL (Universal Resource Locator) del sitio específico de INTERNET, y lo identifica unívocamente en el contexto de la red. Un URL tiene la forma de:

http://www.utn.edu.ec

De donde se extrae el protocolo, el nombre la máquina servidora, y por último el dominio de esa máquina. Así una dirección de correo para este dominio sería alguien@utn.edu.ec. Los nombre de dominio no están reglamentados, sin embargo por lo general éstos finalizan con un código de

dos letras que identifican al país en el que se encuentra el dominio, su omisión significa que está ubicado en EE.UU.

El SMTP hace uso de los dominios para transferir los mensajes, pero para conocer la dirección de red de un dominio dado, usa los servicios de un DNS o sistema de nombres de dominio; que convierte un nombre de dominio dado en una dirección IP

El Modelo SMTP

Como consecuencia de la solicitud de un cliente de correo, a su mail-server, del envío de un mensaje, el mail-server se transforma en un emisor SMTP el cual establece una conexión duplex integral con el receptor SMTP, el cual puede ser la dirección de destino o un host en el camino intermedio hacia éste. El emisor y receptor intercambian mensajes y respuestas en un diálogo del tipo parada y espera; los comandos enviados por el emisor se verán con detalle más adelante así como las respuestas a estos comandos.

Estos comandos tienen la forma de cuatro caracteres ASCII y cuando es necesario uno o más parámetros, también en la forma de caracteres ASCII; tanto los comandos como las respuestas finalizan con la combinación de caracteres especiales <CR/LF> . Además se proporciona un código de respuesta de tres dígitos decimales. También existe la posibilidad de enviar comandos que contengan múltiples líneas de parámetro; por ejemplo el comando DATA, que indica que a continuación se enviará el texto del mensaje, es un comando de líneas múltiples, se delimita estos mensajes con una secuencia <CR/LF> . <CR/LF>

Procedimientos SMTP.-

Establecimiento y Liberación de la conexión.- Una vez abierto el canal de transmisión, los hosts conectados hacen un intercambio de información para asegurarse, que están hablando con quien ellos quieren. Para esto el emisor envía un comando HELO seguido de su dominio. Para finalizar la conexión simplemente el emisor envía el comando QUIT y se libera la conexión.

Transferencia de Mail.- La transferencia de mail tiene tres pasos necesarios cada uno con un comando específico, y con respuestas afirmativas o negativas para cada uno de ellos.

El primer paso es el envío del comando MAIL especificando el origen del mensaje con el "camino inverso", que se usará para reportar errores si los hubiera, el host receptor puede tanto aceptar el mensaje entrante, con una respuesta Positiva (250 OK), como rechazarlo con una respuesta negativa. Este comando indica al receptor el inicio de la transacción de mail por lo que éste debe poner en cero sus tablas de estado, buffers, etc., este comando resetea al receptor. El camino inverso, es la ruta, lista de hosts, que ha seguido el mensaje hasta el host emisor, y tiene a éste al principio de la lista.

El segundo paso comienza con el envío del comando RCPT que indica a quien está destinado el mensaje, con un "camino directo" que indica la ruta que siguió el mensaje hasta el receptor y éste encabeza la lista de hosts del camino. Por simplicidad en este punto se supone que sólo puede conocer al destinatario si es un usuario local, en cuyo caso acepta el mensaje para él (250 OK), si no conoce ese usuario entonces responde negativamente al comando del emisor (550). Luego se verá que éstas no son las únicas posibilidades que existen. Este comando debe repetirse la cantidad de veces necesarias para que el emisor envíe todos los mensajes que tiene para ese dominio.

El último paso de la transacción es el comando DATA, si el receptor acepta envía una respuesta 354, indicando que está listo para recibir el mensaje. Las líneas del mensaje se envían secuencialmente y se marca el final con una línea conteniendo sólo un punto, es decir la secuencia <CR/LF> . <CR/LF>, se usa un método de relleno de caracteres para prevenir la aparición de esta secuencia dentro del texto, es decir si en el cuerpo del mensaje el emisor verifica que el primer carácter de una línea es un punto "." entonces agrega una adicional para que no se malinterprete como un final de texto. En el receptor se ejecuta el proceso inverso, inspecciona cada línea del mensaje si encuentra sólo un punto sabe que es el fin del mensaje,

si encuentra un punto seguido de otros caracteres en la misma línea, elimina el punto que agregó el emisor. Al detectar el final del mensaje el receptor envía al emisor una respuesta 250 OK si todo anduvo bien y responde negativamente si no contaba con los recursos necesarios para almacenar el mensaje.

Re-envio (Forwarding).- En algunos casos la información del destino es incorrecta, pero el host receptor conoce la verdadera dirección del destinatario. Si este es el caso puede tomar dos acciones; o tomar el mensaje y él mismo re-enviarlo, o informar la dirección de destino correcta y rechazar el mensaje. Ambas acciones se informan con una respuesta al comando RCPT, ahora debe quedar claro que el host no solo conoce a sus usuarios locales. Ambas respuestas entregan al emisor la dirección correcta para su uso futuro.

Listas de Correo.- Las listas de correo son tablas en el hosts conteniendo pares de nombres de usuario, casilla de correo, éstos son todos los usuarios que el host reconoce pueden ser locales o remotos. El comando VRFY que tiene como parámetro una cadena de caracteres, que indica el nombre de usuario que se está buscando, y responde el nombre completo del usuario y su casilla de correo, si lo encuentra en su tabla.

Casillas de correo y Terminales.- En la época de publicación del RFC 821 era muy común que los host tuvieran terminales conectadas a ellos por eso, el protocolo provee la posibilidad de enviar mensaje a la casilla de correo (mailing) o enviarlos directamente a la terminal (sending). Para implementar esta distinción se proveen tres comandos que involucran esta emisión de mensajes.

El comando SEND envía un mensaje a una terminal si ésta está activa, en caso contrario devuelve una respuesta negativa 450.

SOML significa **Send OR Mail**, que funciona igual al anterior pero si la terminal no está activa el mensaje se guarda en su casilla de correo.

SAML **Send And Mail** lleva a cavo las dos acciones, un **send** si es posible y un **mail** en cualquier caso.

Re-transmisión.- Cuando llega un mensaje a un host se indica su camino inverso (cómo llegó hasta aquí) y su camino directo (cómo llegar a su destino), el primer elemento del camino inverso debe contener el domino del host emisor del mensaje, y el primer elemento del camino directo deber ser el host receptor de el mensaje actual, en cada retransmisión exitosa del mensaje, el receptor elimina su dominio del camino directo y lo anexa al camino inverso, esto significa que el mensaje pasó por aquí, o por lo menos eso va a intentar, ahora el receptor del mensaje pasa a ser emisor y se conecta con el próximo host del camino directo, si la transmisión tiene éxito el host receptor repetirá el procedimiento hasta llegar al host destino, el último del camino directo. En caso de no tener éxito con la re-transmisión del mensaje, el host debe informar al emisor original del mensaje sobre las causas de la falla, y para eso utiliza la información contenida en el camino inverso, y construye un mensaje de "mail inentregable". Este mensaje se envía con un emisor nulo (MAIL FROM : <>) para así evitar notificaciones de notificaciones, si la notificación no llegó no es tan importante y se previenen potenciales ciclos de notificaciones. Un ejemplo de notificación se presenta a continuación, las razones para que un host no acepte un mensaje pueden ser varias y se desprenden de los comandos RCPT o MAIL.

Cambio de Roles.- En muchas implementaciones es útil que los procesos intercambien los roles de emisor y receptor, para hacer esto el emisor envía un comando TURN, el receptor está en libertad de aceptar (respuesta 250), y pasar a ser el emisor; o rechazar la propuesta (respuesta 502). Este comando es especialmente útil cuando el costo de establecer la conexión es alto, por ejemplo cuando se usa la red pública de teléfonos como canal de transmisión.

Comando del SMTP.- La siguiente es una lista de los comandos del protocolo SMTP con sus parámetros y sintaxis correcta. Los códigos <SP> y <CRLF> significan un espacio en blanco y un retorno de carro, respectivamente.

HELO <SP> <dominio> <CRLF>
MAIL <SP> FROM:<camino-inverso> <CRLF>
RCPT <SP> TO:<camino-directo> <CRLF>
DATA <CRLF>
RSET <CRLF>
SEND <SP> FROM:<camino-inverso> <CRLF>
SOML <SP> FROM:<camino-inverso> <CRLF>
SAML <SP> FROM:<camino-inverso> <CRLF>
VRFY <SP> <cadena> <CRLF>
EXPN <SP> <cadena> <CRLF>
HELP [<SP> <cadena>] <CRLF>
NOOP <CRLF>
QUIT <CRLF>
TURN <CRLF>

Códigos de Respuestas del SMTP.-

211 System status, or system help reply
214 Help message
220 <domain> Service ready
221 <domain> Service closing transmission channel
250 Requested mail action okay, completed
251 User not local; will forward to <forward-path>
354 Start mail input; end with <CRLF>.<CRLF>
421 <domain> Service not available,
closing transmission channel
450 Requested mail action not taken: mailbox unavailable
[E.g., mailbox busy]
451 Requested action aborted: local error in processing
452 Requested action not taken: insufficient system storage
500 Syntax error, command unrecognized
[This may include errors such as command line too long]
501 Syntax error in parameters or arguments
502 Command not implemented
503 Bad sequence of commands
504 Command parameter not implemented
550 Requested action not taken: mailbox unavailable
[E.g., mailbox not found, no access]
551 User not local; please try <forward-path>
552 Requested mail action aborted: exceeded storage
553 Requested action not taken: mailbox name not allowed

554 Transaction failed

Servicio de Transporte.- Al implemetar el SMTP sobre los servicios del TCP se debe establecer una conexión entre un puerto x en el emisor y el puerto 25 del receptor. El protocolo ya tiene asignado este puerto para las conexiones en TCP. De esta manera el SMTP está escuchando el puerto 25 y cuando la conexión está establecida envía la respuesta 220.

TCP soporta la transmisión de bytes de 8 bits, mientras que SMTP transmite caracteres de 7 bits, para hacer transparente esta conversión el bit más significativo se establece en cero.

2.7.6. POP3 (Protocolo de Oficina Postal).

El protocolo de oficina postal fue diseñado para trabajar conjuntamente con el protocolo TCP, inicialmente el proceso está escuchando el puerto 110, a la espera de una conexión, cuando esta se establece el servidor envía un saludo y luego comienza un diálogo en el que se intercambian comandos y respuestas, hasta que la conexión se libera. El POP3 va cambiando entre 3 distintos estados a lo largo de su vida, dependiendo de los resultados de algunos comandos especiales. Los comandos POP3 consisten de 3 o 4 caracteres, con ninguno y algunos parámetros separados por un espacio en blanco. Cada comando finaliza con el par <CRLF>.

Las respuestas muestran el estado del comando, puede ser positivo (+OK) y negativo (-ERR), y además ser seguido por algún tipo de información adicional. En las respuestas multi-línea cada línea enviada termina con el par <CRLF> y la última línea de la transmisión debe ir seguida de un punto "." y el par <CRLF>. Cualquier ocurrencia de esta secuencia en el texto de la respuesta generará un relleno de ese carácter del mismo modo que en el SMTP.

Los estados del POP3 son, Autorización en el que se entra cuando se establece la conexión TCP y sirve para que los usuarios se identifiquen ante

el protocolo. Se entra en el estado de Transacción cuando se hace un identificación positiva del usuario que quiere ingresar, aquí los mensajes pasan del servidor al cliente, un vez finalizado esto, se pasa al estado Actualización, donde elimina los mensajes que el usuario recibió, y así finaliza la conexión y se libera.

El Protocolo de correo- Versión 3 (POP3) fue creado para permitir a una red el acceso dinámico a una casilla sobre un servidor HOST de manera útil. Usualmente, esto significa que el protocolo POP3 es usado para dejar que una red recupere el correo que el servidor accionaría por él. POP3 no esta echo para proveer operaciones de manipulación extensivas de correo en el servidor; normalmente, el correo es bajado y es borrado.

Estado de Autorización.- Una vez que se establece la conexión TCP el host servidor envía una respuesta positiva como una bienvenida. Por ejemplo

+OK POP3 server ready

Se entra así al modo de autorización; el usuario tiene dos maneras de identificarse con el juegos de comandos USER, PASS que especifican en nombre de usuario y su contraseña como parámetros respectivamente. Y con el comando APOP que envía ambos parámetros al mismo tiempo con la diferencia que la contraseña viaja encriptada mediante el algoritmo MD5. Era un tanto peligroso tener viajando por la red el nombre de usuario y la contraseña sin ningún tipo de seguridad. Cualquiera de los dos métodos con una respuesta positiva hacen entrar al protocolo en el estado de transacción, previo un bloqueo del buzón, para asegurar la consistencia de los datos.

Estado De Transacción.- Al entrar en este estado se abre el buzón y se numera a cada mensaje con números decimales comenzando por el 1 y registrando el tamaño en bytes de cada uno. Ahora son posibles los comandos descriptos a continuación, no hay ni una cantidad ni una secuencia especifica para la ejecución de estos comandos, a excepción del comando QUIT que es el que finaliza el estado de transacción:

STAT : no posee argumentos y la respuesta es la cantidad de mensajes actuales en el buzón que no están marcados como eliminados y su tamaño en bytes.

Cliente STAT
Servidor: +OK 2 320

LIST : tiene como parámetro opcional el número de mensaje. Sin parámetro hace que el servidor responda dando la lista de los mensajes no marcados como eliminados en el buzón, junto con el tamaño en bytes de cada uno de ellos. Si se da como parámetro un mensaje se muestra la información de ese mensaje en particular

Cliente: LIST
Servidor: +OK 2 messages (320 bytes)
Servidor: 1 120
Servidor: 2 200
Servidor: .
...
Cliente: LIST 2
Servidor: +OK 2 200
...
Cliente: LIST 3
Servidor: -ERR no such message, only 2 messages in maildrop

RETR : tiene como parámetro obligatorio el número de mensaje. Y devuelve una respuesta multi-línea donde la primera indica, si el mensaje existe, el tamaño y a continuación envía el mensaje.

Cliente: RETR 1
Servidor: +OK 120 Bytes
Servidor: <El servidor POP3 envía todo el mensaje completo>
Servidor: .

DELE : tiene como parámetro el número de mensaje. Este número de mensaje es el que se va a marcar como eliminado.

Cliente: DELE 1
Servidor: +OK message 1 deleted
...
Cliente: DELE 2
Servidor: -ERR message 2 already deleted

NOOP: No tiene argumentos, el servidor POP solamente contesta con una respuesta positiva

C: NOOP
S: + OK

RSET : No tiene parámetros y su acción es desmarcar los mensajes que fueron marcados para su eliminación durante esta transacción.

QUIT : Pasa al próximo estado.

Estado de Actualización.- El estado de actualización desaloja los mensajes marcados como eliminados en el estado de transacción, esto lo hará sólo si el cliente emite un comando QUIT en este estado en caso contrario los mensajes quedarán marcados como eliminados pero no desalojados de el almacenamiento. Además esta comando libera la conexión TCP e informa el estado actual del buzón.

Comandos Adicionales.- El POP3 posee además un par de comandos especiales que no son necesarios para su operación básica. Estos comandos son:

TOP: requiere dos parámetros un identificador de mensaje y un número no negativo que indica la cantidad de líneas que se deben enviar de ese mensaje.

Cliente: TOP 1 10
Servidor: +OK
Servidor: <El POP3 envía las 10 primeras líneas del mensaje 1>
Servidor: .

Ó

Cliente: TOP 100 3
Servidor: -ERR no such message

UIDL: tiene como parámetro opcional un número de mensaje, y devuelve el **id** único de el mensaje que se dio como parámetro, o el de todos los mensajes de el buzón. Este **id** único identifica unívocamente

al mensaje dentro del servidor y es asignado por éste. Los mensajes marcados como eliminados no se listan.

La ventaja principal que tiene este protocolo es que carpetas, mensajes, etc. se guardan en nuestro computador, con lo que nos permite leer el correo recibido sin estar conectado a la red. Además, al leer los mensajes y bajarlos a nuestro ordenador, liberamos espacio en nuestro buzón del Host, con lo cual tenemos menos probabilidades que por descuido se nos llene el buzón y no podamos recibir más mensajes. Es el más extendido (prácticamente todos los programas de correo lo soportan) y es el ideal para conectarse siempre desde un mismo ordenador.

2.7.7. IMAP.

IMAP : Este protocolo es similar al protocolo POP pero sus diferencias radican en la forma en que almacena la información así de como se recupera el e-mail del servidor.

La principal diferencia que encontramos respecto al anterior protocolo es que tanto los mensajes como las carpetas se guardan en el Host. Esto, que puede parecer un inconveniente, es muy útil para conectarse desde ordenadores compartidos, ya que los mensajes no pueden ser leídos por terceras personas, al no quedarse en el PC, además, si no tenemos la posibilidad de conectarnos siempre del mismo ordenador, conseguimos siempre acceder a la totalidad de nuestros mensajes. Hay que tener la precaución de ir borrándolos de vez en cuando para no sobrepasar el límite de capacidad de nuestro buzón. En cuanto al soporte de este protocolo, aunque son pocos los programas de correo que lo soportan por ahora, tenemos la suerte que los dos navegadores más extendidos (Netscape e Internet Explorer -Con Outlook Express u Outlook 98-) sí que pueden trabajar con él.

2.7.8. MIME

Las extensiones MIME para el SMTP no hacen más que complementarlo para hacer más flexible su uso con tipos de datos no-ASCII, MIME especifica tres campos que se incluyen en la cabecera del mensaje, estos son:

- MIME-Version.- que especifica que versión del MIME se usó para codificar ese mensaje;
- Content-Type.- que especifica el tipo y subtipo de los datos no-ASCII; Los valores legales para este campo son: Text, Image, Audio, Video, iApplication, Multipart y Message.
- Content-Transfer-Encoding.- que especifica el tipo de codificación usado para traducir los datos en ASCII. Los valores definidos para este campo son: 7bit, 8bit, binary, quoted-printable, base64, itef-token, x-token

Los tipos del campo contenido, casi hablan por si mismos a excepción de uno, el tipo multipart; que tiene 4 subtipos, mixed, alternative, parallel y digest. Este tipo significa que el mensaje contiene en sí mismo información de diferentes tipos o en distintos formatos. Cada una de las partes del mensaje se separa con un delimitador, que toma la forma de una cadena especificada en el campo Boudary que sigue al Content-Type. El subtipo Mixed indica que el mensaje encierra parte de distintos tipos, en cada comienzo de una nueva parte, después de la cadena delimitadora, debe especificarse el tipo y la codificación de la parte. El subtipo Alternative permite que el mismo mensaje se codifique usando distintos métodos, para asegurarse que el mensaje pueda ser leído por el programa del destinatario. El subtipo Parallel indica que las partes deben mostrarse juntas. Y por último el subtipo Digest indica que contiene un conjunto de mensaje, por ejemplo una discusión por e-mail.

RECOMENDACIONES:

Cada protocolo tiene sus propios estándares publicados en Internet (http://www.rfc-editor.org), se puede profundizar en el tema de un protocolo especifico e inclusive se puede llegar a implementarlo siguiendo los estándares.

BIBLIOGRAFIA:

Ambegaonkar Prakash. Kit de Recursos de Intranet
Editorial Osborne / McGraw-Hill primera edición

Feit TCP/IP
Editorial McGraw-Hill, primera edición 1998

Greer Tyson.- Así son las Intranets
Editorial Microsoft Press primera edición 1997

Kris Jamsa / Ken Cope. Programación en Internet (Curso sobre TCP/IP)
Editorial McGraw-Hill primera edición 1996

Sheldon Tom.- LAN TIMES Guía de Interoperabilidad
Editorial Osborne / McGraw-Hill primera edición 1996 .

www.rfc-editor.org Sitio en el cual se encuentran los documentos RFC

www.w3c.org W3 Consortium Organization.