*Final De Redes y Comunicaciones*

***UNIDAD 1 – Introducción***

***¿Qué es una Red?***

Una Red es un grupo de computadoras / dispositivos interconectados, que comparten información (archivos), recursos (CD, impresoras, etc.) y servicios (accesos a internet, e-mail, chat, juegos, etc.).

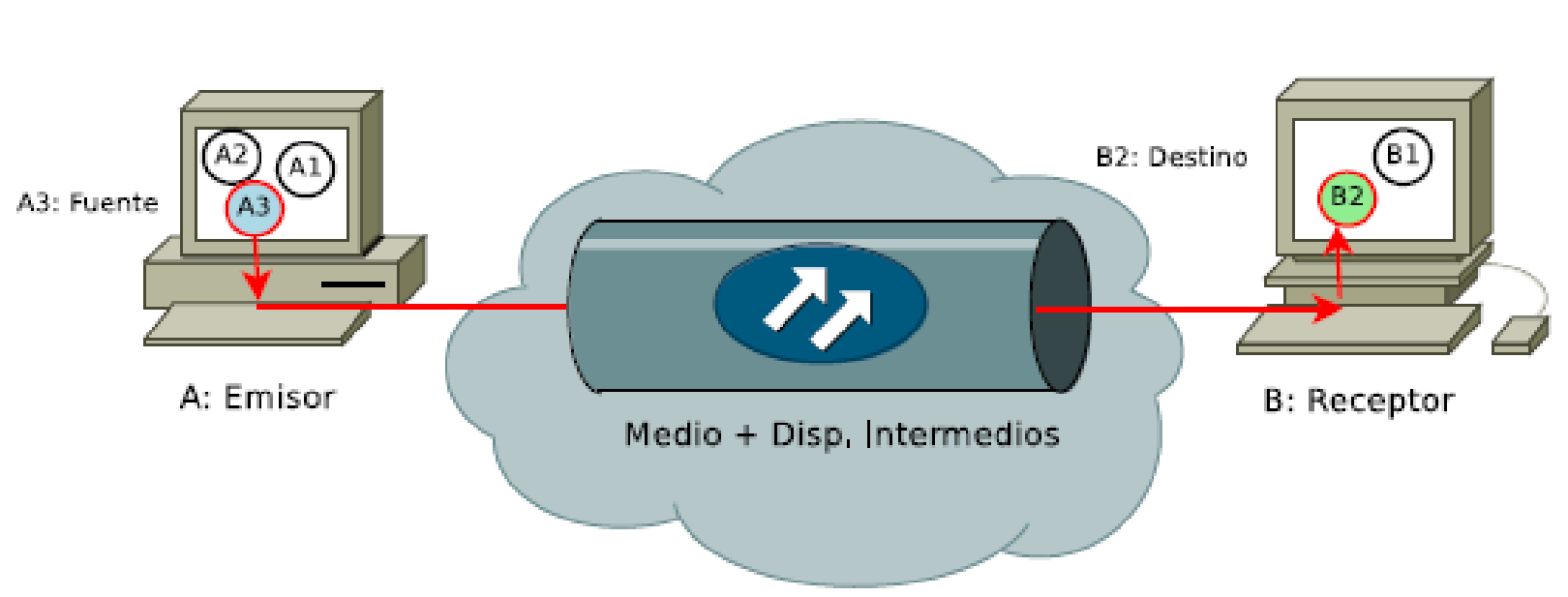
El Objetivo es compartir archivos, intercambiar información y recursos.

El conjunto, computadoras, software de red, medios y dispositivos de interconexión, forma un sistema de comunicación.

Como en todo proceso de [comunicación](http://es.wikipedia.org/wiki/Comunicaci%C3%B3n) se requiere de un [emisor](http://es.wikipedia.org/wiki/Emisor), un [mensaje](http://es.wikipedia.org/wiki/Mensaje), un [medio](http://es.wikipedia.org/wiki/Medio_de_comunicaci%C3%B3n) y un [receptor](http://es.wikipedia.org/wiki/Receptor_(comunicaci%C3%B3n)). La finalidad principal para la creación de una red de computadoras es compartir los recursos y la información en la distancia, asegurar la [confiabilidad](http://es.wikipedia.org/wiki/Integridad_de_datos) y la disponibilidad de la información, aumentar la [velocidad de transmisión de los datos](http://es.wikipedia.org/wiki/Velocidad_de_conexi%C3%B3n) y reducir el costo general de estas acciones. Un ejemplo es [Internet](http://es.wikipedia.org/wiki/Internet), la cual es una gran red de millones de computadoras ubicadas en distintos puntos del planeta interconectadas básicamente para compartir información y recursos.

***Componentes de un sistema de Comunicación***

* Fuente (Software).
* Emisor/Transmisor (Hardware).
* Medio de transmisión y dispositivos intermedios (Hardware).
* Procesos intermedios que tratan la información (Software y Hardware).
* Receptor (Hardware).
* Destino (Software).
* Otros: Protocolos (Software), Información, mensaje transmitido (Software).
* Señal de Información, materialización del mensaje sobre el medio (Hardware?).



***¿Qué es Internet?***

Internet es una red de computadoras mundiales descentralizada, pública, esta es, una red que conecta millones de dispositivos de cómputos a través del mundo. La mayoría de los dispositivos son las PC tradicionales, las estaciones de trabajo basado en UNIX, y los llamados servidores, que almacenan y transportan información del tipo páginas web y mensajería de correo electrónico.

En la jerga, todos los adaptadores se llaman host y sistemas terminales. Se conectan mediante enlaces de comunicación.

La tasa de transmisión de enlace se llama ancho de banda, y se mide normalmente en bits/segundo.

Los sistemas terminales normalmente no se conectan directamente entre sí mediante enlaces sencillos de comunicación, sino que se conectan unos a los otros a través de dispositivos de comunicación llamados **routers**. (Opera en la capa 3, nivel de red) Este es un dispositivo para la interconexión de redes, que permite asegurar el enrutamiento de paquetes entre redes o determinar la ruta que debe tomar el paquete de datos.

Un router toma un trozo de información de los enlaces de comunicación de entrada y lo difunde en uno de los enlaces de comunicación de salida. Este trozo de información se lo denomina **Paquete**. El recorrido que el paquete realiza desde el sistema final que lo envía al sistema final receptor se denomina **ruta o recorrido** través de la red.

Internet utiliza una técnica conocida como **Conmutación de paquetes**, que permite que múltiples sistemas terminales que se comunican compartan al mismo tiempo un recorrido o parte de uno.

Los Sistemas terminales acceden a internet a través de **proveedores de servicios de internet (ISP)**, Cada IPS es una red de routers y enlaces de comunicación. Proporcionan diferentes tipos de acceso a la red para los sistemas terminales, incluyendo acceso a través banda ancha, redes de alta velocidad, y acceso sin cable.

Los sistemas terminales, routers, y otros elementos de internet, ejecutan **Protocolos** que controlan el envío y recepción de información en internet (TCP/IP). **TCP** (protocolo de control de transmisión), **IP** (protocolo internet), Son los dos protocolos más importantes de internet. El protocolo IP especifica el formato de los paquetes que son enviados y recibidos entre routers y sistemas terminales.

La estructura y el modo de funcionamiento de las redes informáticas actuales están definidos en varios [estándares](http://es.wikipedia.org/wiki/Norma_(tecnolog%C3%ADa)), siendo el más importante y extendido de todos ellos el modelo [TCP/IP](http://es.wikipedia.org/wiki/TCP/IP) basado en el modelo de referencia [OSI](http://es.wikipedia.org/wiki/Modelo_OSI). Este último, estructura cada red en siete capas con funciones concretas pero relacionadas entre sí; en TCP/IP se reducen a cuatro capas. Existen multitud de protocolos repartidos por cada capa, los cuales también están regidos por sus respectivos estándares.

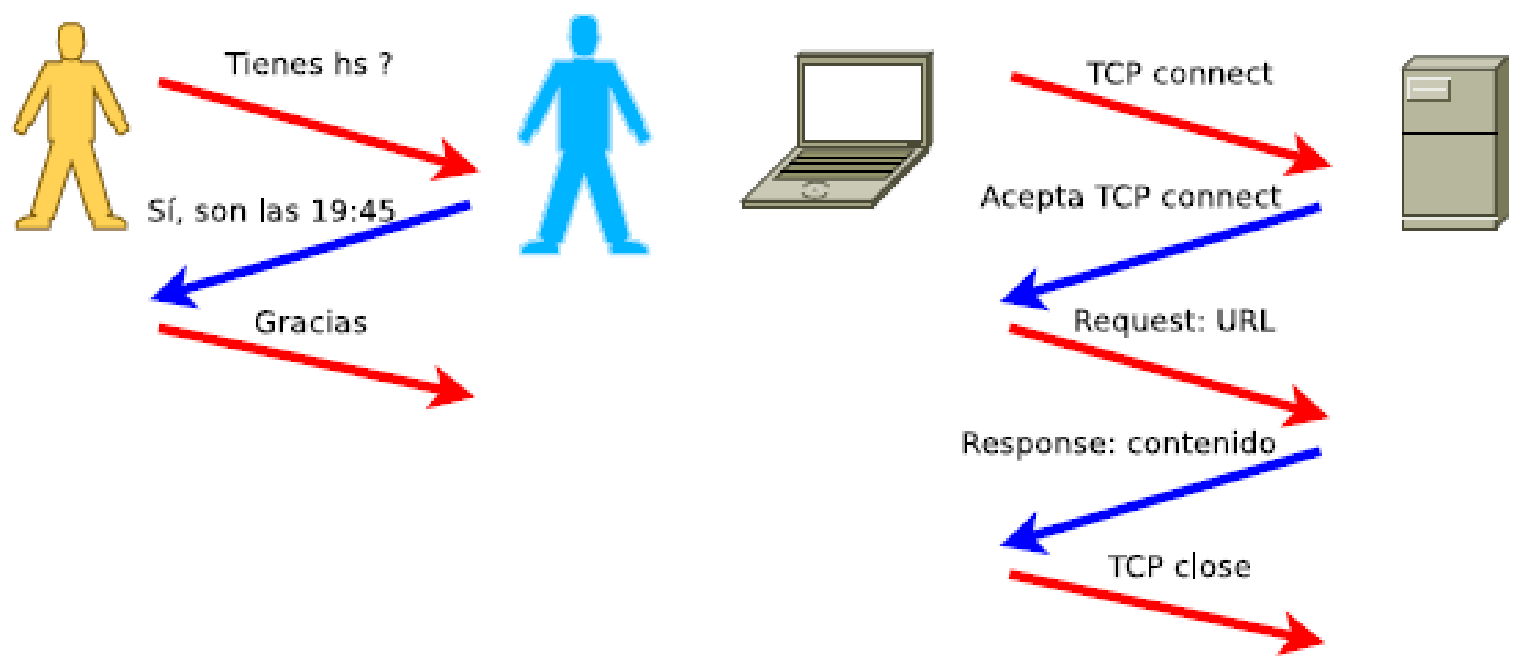
***Componentes de una Red***

* Computadoras, en el modelo de Internet: Hosts (PCs, laptops, servidores).
* Routers/switches.
* Gateways.
* Access points.
* NIC (placas de red), Modems.
* Vínculos/enlaces: conformados por:

◦ Medios: cables, fibras ópticas, señales electromagnéticas, antenas, interfaces, etc.

* Etc... , las componentes de la red deben interactuar y combinarse a través de reglas.

**Protocolo:** El conjunto de conductas y normas a conocer, respetar y cumplir no solo en el medio oficial ya establecido, sino también en el medio social, laboral, etc. (Formato y orden de los msj y acciones)



***Dos máquinas con distintos sistemas operativos pueden formar parte de la misma red porque ambas máquinas utilizan los mismos protocolos.***

Un protocolo define el formato, el orden de los mensajes intercambiados y las acciones que se llevan a cabo en la transmisión y/o recepción de un mensaje u otro evento.

**Protocolo de Red:** conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de una red. Permiten la comunicación y están implementados en las componentes.

Se trata de las reglas o el estándar que define la [sintaxis](http://es.wikipedia.org/wiki/Sintaxis), [semántica](http://es.wikipedia.org/wiki/Sem%C3%A1ntica) y [sincronización](http://es.wikipedia.org/wiki/Sincronizaci%C3%B3n) de la [comunicación](http://es.wikipedia.org/wiki/Comunicaci%C3%B3n), así como también los posibles [métodos de recuperación de errores](http://es.wikipedia.org/wiki/C%C3%B3digos_detectores_y_correctores_de_error). Los protocolos pueden ser implementados por [hardware](http://es.wikipedia.org/wiki/Hardware), por [software](http://es.wikipedia.org/wiki/Software), o por una combinación de ambos.

Un protocolo de comunicación, también llamado en este caso protocolo de red, define la forma en la que los distintos mensajes o [tramas](http://es.wikipedia.org/wiki/Trama_de_red) de bit circulan en una red de computadoras.

***Protocolos de Redes Propietarios***

* A principios de los 80’, las compañías comenzaron a implementar redes propias (privadas y cerradas).
* Primeras Redes Propietarias.
* Consecuencia: Cada red tenía sus especificaciones propias (protocolos).
* Resultados: Incompatibilidad. La comunicación entre redes era muy difícil, evolución más lenta, carencia de estándares.
* Complejidad de modelos.

***Combinación de Protocolos***

* La cantidad de componentes de red a interactuar genera complejidad, se requiere una organización de las mismas.
* Se requieren **Modelos de Organización**.
* Modelo en Capas: Layering, divide la complejidad en componentes reusables.
  + Reduce complejidad en componente más pequeñas.
  + Las capas de **abajo ocultan la complejidad** a las de arriba, **abstracción**.
  + Las capas de arriba utilizan servicios de las de abajo: **Interfaces**, similar a APIs.
  + Los **cambios en una capa no deberían afectar a las demás** si la interfaz se mantiene.
  + Facilita el desarrollo, evolución de las componentes de red asegurando interoperabilidad.
  + Facilita aprendizaje, diseño y administración de las redes.

***Internet Ofrece dos tipos de servicios:***

* **Servicios orientados a la conexión (TCP):** garantiza que los datos transmitidos de un emisor hasta el receptor serán entregados eventualmente al receptor completamente y en orden.
* **Servicios no orientados a la conexión (UDP):** no proporciona ninguna garantía sobre eventual entrega y/o recepción de un mensaje u otro evento.

***RFC (Request for Comments)***

Son notas sobre internet, Es un documento cuyo contenido es una propuesta oficial para un nuevo protocolo de la red de Internet, que se explica con todo detalle para que en caso de ser aceptado pueda ser implementado sin ambigüedades.

Cada RFC tiene un título y un número asignado, que no puede repetirse ni eliminarse aunque el documento quede obsoleto.

**Categorías**: informativos, propuestas de estándares nuevos o históricos.

La familia de protocolos de Internet está todavía evolucionando mediante el mecanismo de Petición de Comentario (RFC). Los nuevos protocolos (la mayoría de los protocolos de aplicación) los han diseñado e implementado investigadores y científicos y han sido expuestos a la comunidad de Internet en forma de RFC. El Internet Architecture Board (IAB) supervisa el mecanismo RFC. La mayor fuente de RFCs es la Internet Engineering Task Force (IETF) que es un subsidiario del IAB. Sin embargo, cualquiera puede proponer una memorándum como un RFC al editor de RFC. Existe una serie de reglas que los autores de RFC deben seguir para que se acepten. Estas reglas se describen en un RFC (RFC 1543) que indica cómo considerar una propuesta para un RFC.

Una vez que se ha publicado un RFC, todas las revisiones y suplementos se publicarán como nuevos RFCs. Un nuevo RFC que revise o reemplace uno existente se dice "actualizado" u "obsoleto". El RFC existente se dice "actualizado por" u "obsoleto por" el nuevo. Por ejemplo el RFC 1521 que describe el protocolo MIME es una "segunda edición", siendo una revisión del RFC 1341 y el RFC 1590 es una corrección al RFC 1521. RFC 1521 es por tanto etiquetada como "Obsoleto RFC 1341; Actualizado por RFC 1590". Por consiguiente, no existe confusión alguna de si dos personas se están refiriendo a versiones diferentes de un RFC, dado que no hay nunca versiones diferentes.

Algunos RFCs se describen como documentos de información que otros describen como protocolos Internet. El Internet Architecture Board (IAB) mantiene una lista de RFCs que describen la familia de protocolos. Cada uno de estos tiene asignado un estado y un status o categoría.

***Un protocolo de Internet puede tener uno de los siguientes estados:***

* **Estándar**, el IAB ha establecido esto como un protocolo oficial para Internet. Se separan en dos grupos:

1. Protocolo IP y citados, protocolos aplicados enteramente a Internet.

2. Protocolos específicos de red, generalmente especificaciones de cómo hacer IP sobre tipos particulares de redes.

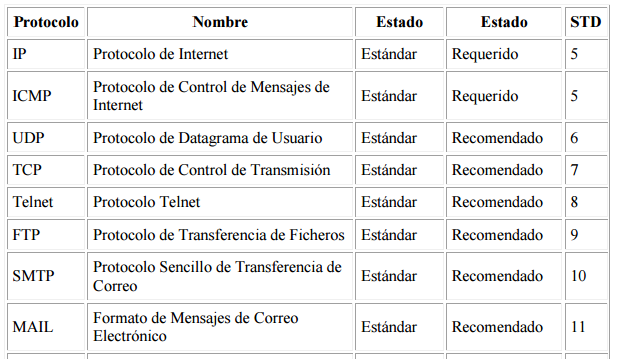
* **Estándar Borrador**, el IAB está considerando activamente este protocolo como un posible protocolo estándar. El IAB somete los comentarios y resultados de pruebas. Existe una posibilidad que cambie, esos cambios serán hechos en un borrador del protocolo antes de liberarlos como estándar.
* **Estándar Propuesto**, estos son protocolos propuestos que debe considerar IAB para su estandarización en el futuro. Son implementaciones deseables y comprobaciones de varios grupos. Es probable la revisión del protocolo.
* **Experimental**, un sistema no debería implementar un protocolo experimental a no ser que esté participando en el experimento y ha coordinado su uso del protocolo con el desarrollador del protocolo.
* **Informativo**, los protocolos desarrollados por otras organizaciones, o vendedores, o que están por otras razones fuera del alcance de IAB deben publicarse como RFCs por conveniencia de la comunidad de Internet como protocolos informativos. Tales protocolos pueden en algunos casos también estar recomendados para uso en Internet por IAB.
* **Histórico**, estos son protocolos que con poca probabilidad llegan a ser estándares en Internet porque los han reemplazado los desarrolladores más tarde o por falta de interés.

***Definiciones de status o categoría del protocolo:***

* **Requerido,**  un sistema debe implementar los protocolos requeridos.
* **Recomendado**, un sistema debe implementar los protocolos recomendados.
* **Electivo,** un sistema puede o no implementar un protocolo electivo. La noción general es que si se va a hacer algo como esto, se debe hacer exactamente esto.
* **Uso limitado,** estos protocolos están para usar en circunstancias limitadas. Esto puede ser debido a su estado experimental, naturaleza específica, funcionalidad limitada o estado histórico.
* **No recomendado,** estos protocolos no se recomiendan para uso general. Esto se puede deber a su funcionalidad limitada, naturaleza específica o estado experimental o histórico.

Cuando un protocolo alcanza el estado de estándar se le asigna un **número estándar (STD)**. El propósito de los números STD es indicar claramente qué RFCs describen los estándares de Internet. Los números STD referencian múltiples RFCs cuando la especificación de un estándar se divide en múltiples documentos. No como con los RFCs, donde el número se refiere a un documento específico, los números STD no cambian cuando un estándar se actualiza. Los números STD, sin embargo, no tienen número de versión dado que todas las actualizaciones se realizan vía RFCs y los números de RFC son únicos. De este modo, para especificar sin ambigüedad qué versión de un estándar único se está refiriendo, se pondría de manifiesto el número estándar y todos los RFCs que incluye. Por ejemplo: el Sistema de Nombres de Dominio (DNS) es STD 13 y se describe en los RFCs 1034 y 1035. Para referenciar el estándar se podría utilizar algo como "STD-13/RFC-1034/RFC-1035". Para una descripción de los Procesos Estándares, ver RFC 1602 -- Los Procesos Estándares de Internet - Revisión 2.

Para dar una idea sobre la importancia de los principales protocolos, se listan algunos de ellos junto con su estado actual y número de STD donde es aplicable en la tabla que se muestra abajo. La lista completa puede encontrarse en el RFC 1780 - Estándares de Protocolos Oficiales de Internet.



**NOTA**

|  |
| --- |
| Hay dos categorías que se pueden clasificar los **sistemas finales o End Systems**:   * **Cliente**: Solicitan y reciben un servicio * **Servidor**: responden a las solicitudes de los clientes   Los clientes suelen ser por lo general computadores personales, mientras que los servidores suelen ser máquinas más potentes que alojan servicios, como dominio de web o correo.  Los sistemas terminales o anfitriones (host) son aquellos dispositivos que alojan programas de aplicación, como navegadores, clientes de correo electrónico o servidores de correo.  Diferencia entre la red de **Telefonía** e **Internet**: La diferencia principal es que la red de telefonía es la red de conmutación de circuitos e internet es una red de conmutación de paquetes.  ***Conmutación de circuitos***: los equipos deben establecer un cambio físico entre los medios de comunicación previa a la conexión entre los usuarios. Este camino permanece activo durante la comunicación entre los usuarios, liberándose al terminar la comunicación.   * + **Ventajas:**     - La transmisión se realiza en tiempo real, siendo adecuado para comunicación de voz y video.     - Acaparamiento de recursos. Los nodos que intervienen en la comunicación disponen en exclusiva del circuito establecido mientras dura la sesión.     - No hay contención. Una vez que se ha establecido el circuito las partes pueden comunicarse a la máxima velocidad que permita el medio, sin compartir el ancho de banda ni el tiempo de uso.     - El circuito es fijo. Dado que se dedica un circuito físico específicamente para esa sesión de comunicación, una vez establecido el circuito no hay pérdidas de tiempo calculando y tomando decisiones de encaminamiento en los nodos intermedios. Cada nodo intermedio tiene una sola ruta para los paquetes entrantes y salientes que pertenecen a una sesión específica.     - Simplicidad en la gestión de los nodos intermedios. Una vez que se ha establecido el circuito físico, no hay que tomar más decisiones para encaminar los datos entre el origen y el destino.   + **Desventajas:**     - Retraso en el inicio de la comunicación. Se necesita un tiempo para realizar la conexión, lo que conlleva un retraso en la transmisión de la información.     - Acaparamiento (bloqueo) de recursos. No se aprovecha el circuito en los instantes de tiempo en que no hay transmisión entre las partes. Se desperdicia ancho de banda mientras las partes no están comunicándose.     - El circuito es fijo. No se reajusta la ruta de comunicación, adaptándola en cada posible instante al camino de menor costo entre los nodos. Una vez que se ha establecido el circuito, no se aprovechan los posibles caminos alternativos con menor coste que puedan surgir durante la sesión.     - Poco tolerante a fallos. Si un nodo intermedio falla, todo el circuito se viene abajo. Hay que volver a establecer conexiones desde el principio.   ***Conmutación de paquetes:*** el emisor divide los mensajes a enviar en un número arbitrario de paquetes del mismo tamaño, donde adjunta una cabecera y una dirección origen y destino, así como datos de control que luego serán transmitidos por diferentes medios de conexión entre nodos temporales hasta llegar a su destino. Optimiza la capacidad de transmisiones a través de las líneas existentes.   * + **Ventajas:**     - Si hay error de comunicación, se retransmite una cantidad de datos aún menor que en el caso de mensajes.     - En caso de error en un paquete, solo se reenvía ese paquete, sin afectar a los demás que llegaron sin error.     - [Comunicación interactiva](http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_interactiva). Al limitar el tamaño máximo del paquete, se asegura que ningún usuario pueda monopolizar una línea de transmisión durante mucho tiempo (microsegundos), por lo que las redes de conmutación de paquetes pueden manejar tráfico interactivo.     - Aumenta la flexibilidad y rentabilidad de la red:       * Se puede alterar sobre la marcha el camino seguido por una comunicación (por ejemplo, en caso de avería de uno o más enrutadores).       * Se pueden asignar prioridades a los paquetes de una determinada comunicación. Así, un nodo puede seleccionar, de entre su cola de paquetes en espera de ser transmitidos, aquellos que tienen mayor prioridad.   + **Desventajas:**     - Mayor complejidad en los equipos de conmutación intermedios, que necesitan mayor velocidad y capacidad de cálculo para determinar la ruta adecuada en cada paquete.     - Duplicidad de paquetes. Si un paquete tarda demasiado en llegar a su destino, el host receptor(destino) no enviará el acuse de recibo al emisor, por el cual el host emisor al no recibir un acuse de recibo por parte del receptor este volverá a retransmitir los últimos paquetes del cual no recibió el acuse, pudiendo haber redundancia de datos.     - Si los cálculos de encaminamiento representan un porcentaje apreciable del tiempo de transmisión, el rendimiento del canal (información útil/información transmitida) disminuye. |

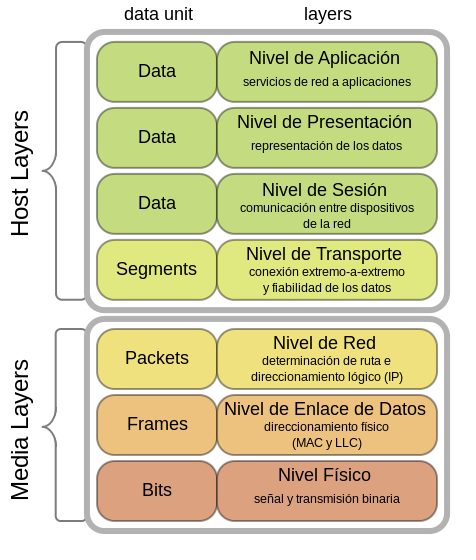
***Modelo OSI***

A mediados de la década de 1980, una de las necesidades que surgió de los sistemas de comunicaciones es el establecimiento de estándares de red, sin ellos sólo podrían comunicarse entre equipos del mismo fabricante y que usaran la misma tecnología. Para enfrentar este problema, la Organización Internacional para la Estandarización (ISO), desarrolló un modelo de red que ayuda a los fabricantes a crear redes que sean compatibles con otras redes.

El modelo OSI no garantiza la comunicación entre equipos pero pone las bases para una mejor estructuración de los protocolos de comunicación. Tampoco existe un sistema de comunicación que los siga estrictamente, siendo la familia de los protocolos TCP/IP la que más se acerca.

Está basado en los modelos de red en capas como DECNET, SNA, TCP/IP. Y es un modelo abierto y estándar. Es un modelo de referencia.

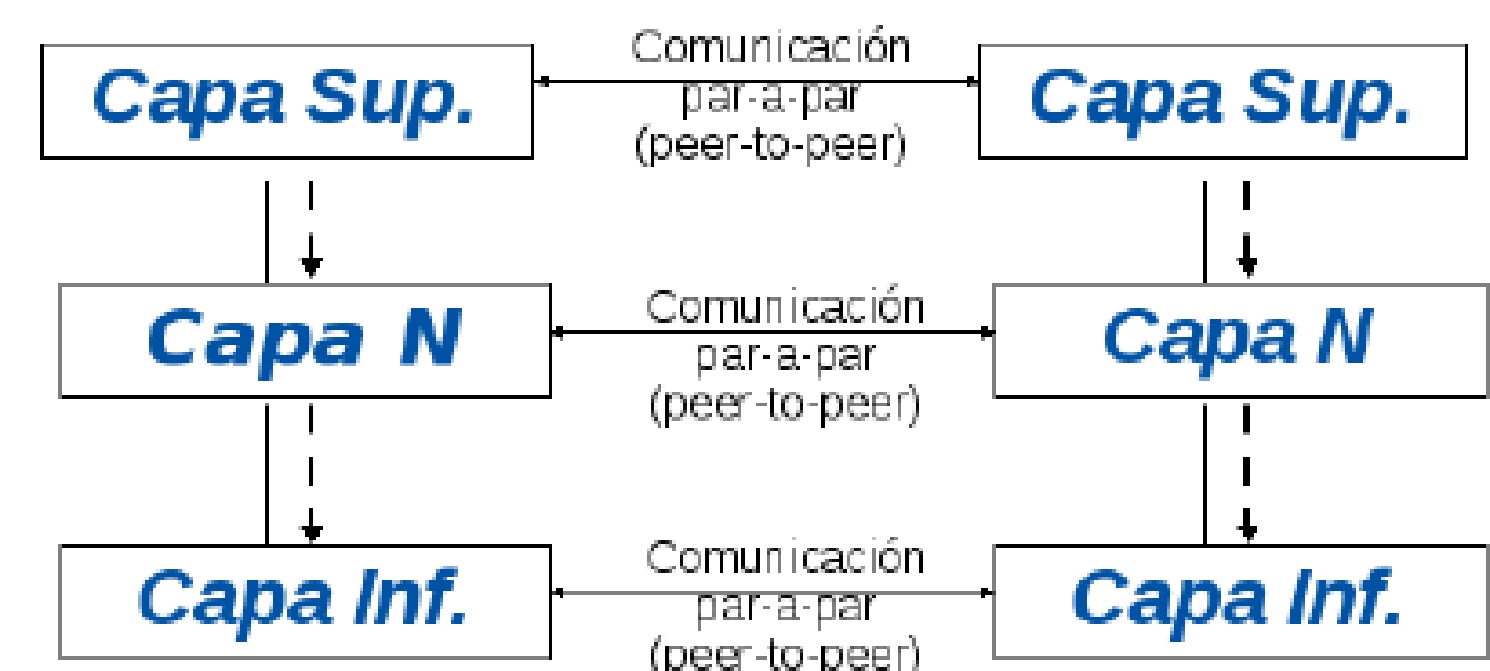
El Modelo OSI describe siete niveles para facilitar los interfaces de conexión entre sistemas abiertos.



**Modelo en Capas:** capa ofrece servicios a la capa superior, usa servicios de la capa inferior, mediante interfaz.

Es mucho más fácil cambiar la implementación del servicio proporcionado por la capa. El resto del sistema no cambia cuando se modifica la implementación de una capa.

Para reducir la complejidad del diseño, los diseñadores de redes organizan protocolos en capas, con una arquitectura de protocolos en capas, cada protocolo pertenece a alguna de las capas.



***Modelo TCP/IP o Protocolo de Internet***

Modelo que se convirtió en estándar. Está basado en el modelo de referencia OSI.

Qué protocolos se encuentran en Internet?

• Modelo Abierto.

• Varios protocolos de nivel de enlace: Ethernet, PPP, Frame-Relay, 802.11ª/b/g, etc. (No definidos por TCP/IP).

• Protocolos propios de Internet y Transporte (Núcleo): ARP, IP, ICMP, TCP, UDP, OSPF, etc.

• Protocolos de Aplicaciones: DNS, HTTP, FTP, SSH, SMTP, etc.

• API abierta para generar nuevos protocolos.

***Capas:***

***Capa De Aplicación:*** esta capa es responsable de soportar las aplicaciones de red. Incluye muchos protocolos, entre ellos HTTP para soportar Web, SMTP para soportar correo electrónico y FTP para soportar transferencias de archivos.

***Capa de Transporte:*** esta capa proporciona el servicio de transporte de mensajes de la capa de aplicación entre los datos del cliente y el servidor de una aplicación. Hay dos protocolos de transportes: UDP y TCP.

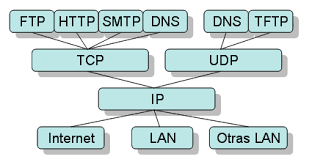
***Capa de Red:*** esta capa es responsable de rutar los datagramas de una maquina a otra. Tiene dos componentes principales. Tiene un protocolo que define los campos del datagrama IP y cómo actúan los sistemas terminales y routers sobre esos campos. Este es el protocolo IP. También contienen protocolos de rutado que determinan las rutas que toman los datagramas entre las fuentes y destinos.

***Capa de Enlace:*** la capa de red debe confiar en los servicios de la capa de enlace. EN partículas, en cada nodo, la capa de red baja el datagrama hacia la capa de enlace, que entrega el datagrama al siguiente nodo de la ruta. En el siguiente nodo, la capa de enlace sube el datagrama a la capa de red.

Los servicios proporcionados por la capa de enlace dependen del protocolo específico de dicha capa que se emplea sobre el enlace.

***Capa Física:*** El trabajo de la capa física es mover los bits individuales del marco de un nodo hasta el siguiente. Los protocoles de esta capa son dependientes del enlace, y dependen además del medio de transmisión actual del enlace.

***Modelo de reloj de arena (hourglass)***



***El PDU de cada capa del Stack de TCP/IP es: (protocol data unit)***

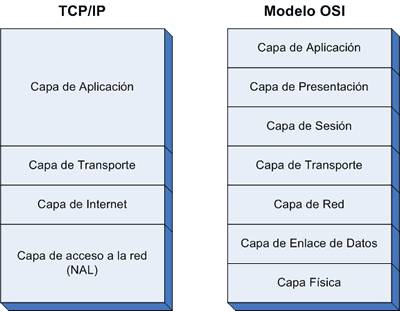
**1. Aplicación:** Mensaje

**2. Transporte:** Segmento

**3. Red:** Datagrama

**4. Enlace:** Marco/Trama

***Comparación entre modelo OSI y modelo TCP/IP***



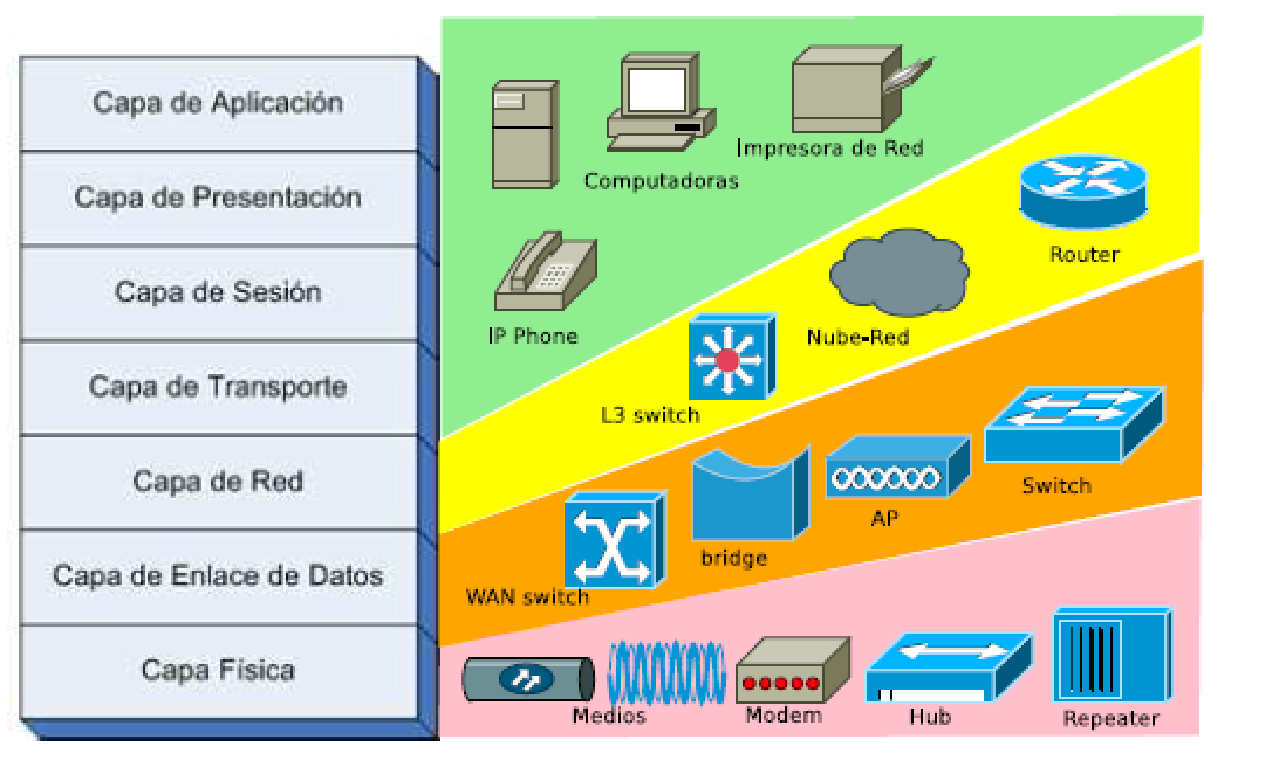
***Similitudes***

* Ambos se dividen en capas.
* Ambos poseen una capa de aplicación aunque ofrecen servicios muy distintos.
* Ambos tienen capas de transporte similares.
* Ambos tienen capa de red similar pero con distinto nombre.
* Ambos modelos utilizan tecnología de conmutación por paquetes y no de conmutación por circuitos.
* Es importante conocer ambos modelos.

***Diferencias***

* TCP/IP combina las funciones de la capa de presentación y de sesión en la capa de aplicación.
* TCP/IP combina la capa de enlace de datos y la capa física del modelo OSI en una sola capa.
* TCP/IP es más simple porque contiene menos capas.
* Los protocolos TCP/IP son los estándares en torno a los cuales se desarrolló Internet. De modo que la credibilidad del modelo TCP/IP se debe en gran parte a sus protocolos.
* El modelo OSI es un modelo más de referencia, teórico, aunque hay implementaciones.

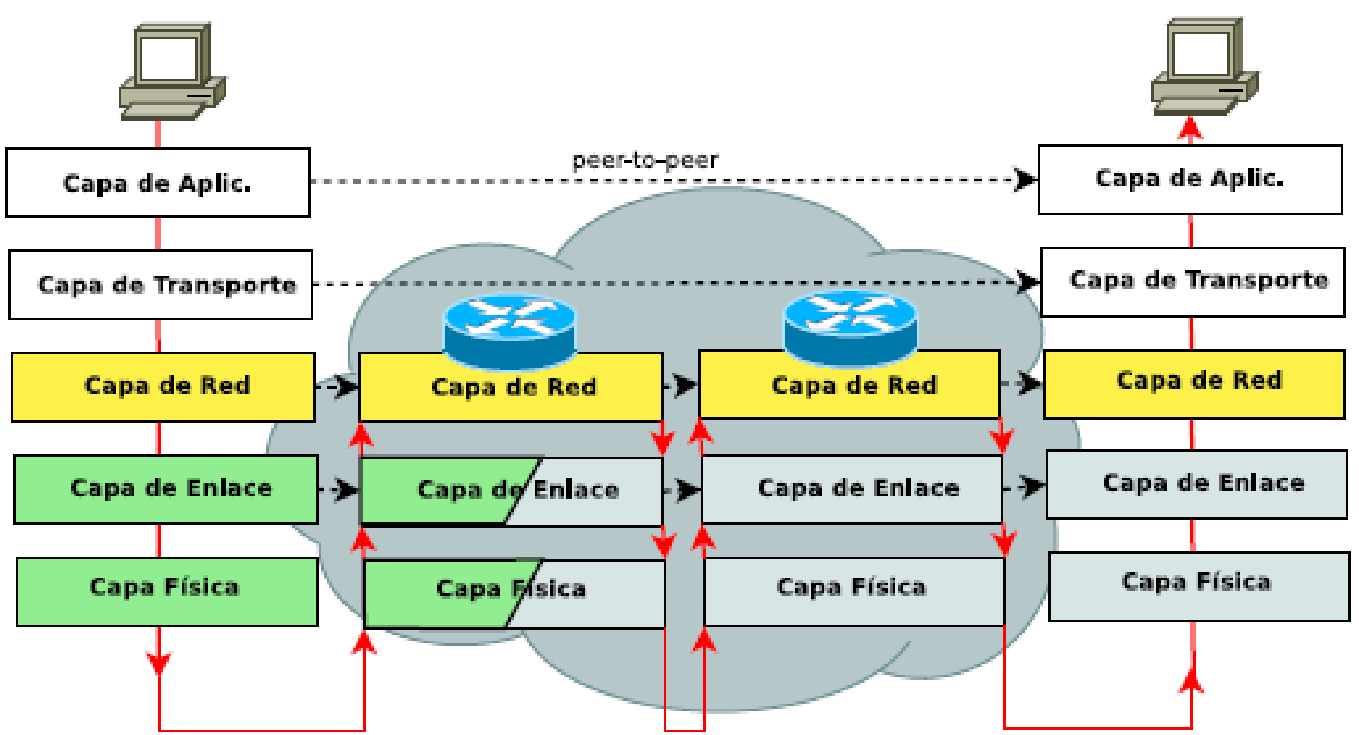
***Dispositivos y capas***



***Comunicación entre capas Peer-to-Peer***

* Cada capa usa el servicio de la de abajo, y cada capa se comunica con la capa del otro extremo.
* Cada capa usa el servicio de la de abajo.

Las tecnologías ‘peer to peer’ (P2P) hacen referencia a un tipo de arquitectura para la comunicación entre aplicaciones que permite a individuos comunicarse y compartir información con otros individuos sin necesidad de un servidor central que facilite la comunicación. Es importante destacar que el término “P2P” se refiere a un tipo de arquitectura de aplicaciones y no a la funcionalidad específica de una aplicación final; es decir, la tecnología P2P es un medio para alcanzar un fin superior. Sin embargo, a menudo se utiliza el término “P2P” como sinónimo de “intercambio de archivos”, ya que éste es uno de los usos más populares de dicha tecnología. No obstante, existen muchos otros usos de la tecnología P2P, por ejemplo Skype utiliza una arquitectura P2P híbrida para ofrecer servicios VoIP, mientras que Tor utiliza una arquitectura P2P para ofrecer una funcionalidad de enrutamiento anónimo. **La ventaja** principal de la tecnología P2P es que saca el máximo partido de los recursos (ancho de banda, capacidad de almacenamiento, etc.) de los muchos clientes/peers para ofrecer servicios de aplicación y red, sin tener que confiar en los recursos de uno o más servidores centrales. De este modo se evita que tales servidores se conviertan en un cuello de botella para toda la red**. Otra ventaja** de la tecnología P2P es que no existe una autoridad central única que se pueda eliminar o bloquear y colapsar toda la red P2P. Esto dota a la red de la capacidad de sobrevivir por sí misma y de una gran robustez.



**AMENAZAS P2P**:

• *Fuga de datos:* Publicación de información o archivos de la empresa de forma consciente o inconsciente.

• *Violación de derechos de propiedad intelectual:* Descarga por parte de los usuarios de contenidos ilegales/protegidos por derechos de propiedad intelectual.

• *Consumo de recursos:* Consumo excesivo de ancho de banda, incluyendo un consumo de ancho de banda adicional por el servicio prestado a otros peers en lugar de para usos directamente relacionados con la actividad del usuario.

• *Control de acceso*: La naturaleza descentralizada de las tecnologías P2P hacen que sea difícil prevenir su uso mediante el empleo de mecanismos tradicionales para el control del acceso a la red.

• *Retención de datos:* Registrar y auditar de forma correcta los datos de las comunicaciones P2P es una tarea difícil y en muchos casos imposible.

• *Malware:* Los usuarios pueden descargar virus, troyanos u otros tipos de malware.

• Pérdida de tiempo: El tiempo que se emplea utilizando las aplicaciones P2P es tiempo que no se dedica a trabajar.

***Acceso residencial:***

Se refiere a la conexión de un sistema terminal de hogar a un router de borde. El modem doméstico convierte la salida digital de la PC en un formato analógico de transmisión sobre la línea telefónica analógica.

Las tecnologías de acceso de banda ancha están proporcionando a los usuarios residenciales tasas de bits más elevadas, y también proporcionan a los usuarios un medio de acceder a Internet y hablar por teléfono al mismo tiempo. Hay dos tipos de acceso residencial de banda ancha: línea de subscriptor digital (DSL) y cable hibrido de fibra y coaxial (HFC).

DLS está proporcionado por una compañía de teléfono. Divide los enlaces de comunicación entre el hogar y el ISP en tres bandas de frecuencias sin solapamiento:

* Un canal de descarga de alta velocidad
* Un canal de descarga de velocidad Media
* Un canal de teléfono ordinario de dos direcciones.

A diferencia de los módems telefónicos han diseñado DSL para distancias cortas entre módems residenciales y el ISP, consiguiendo tasas sustancialmente más altas que las del acceso telefónico.

DSL y los módems telefónicos utilizan las líneas de teléfonos ordinarias, las redes de acceso HFC son extensiones de la red de cable ordinaria utilizada para la difusión de la televisión por cable.

Como DSL, HFC requiere módems especiales, llamados de cable. El módems de cable es un dispositivo externo, y conecta a la PC del hogar mediante un puerto Ethernet 10-baseT.

HFC es que es un medio de difusión compartido.

Ejemplo: Si varios usuarios están descargando diferentes archivos MP3 en el canal de descarga, la tasa a la que cada usuario recibe su MP3 será significativamente más baja que la de descarga.

Como el canal hacia arriba es también compartido, los paquetes enviados por dos hogares diferentes al mismo tiempo colisionarán, lo que disminuirá el ancho de banda efectivo hacia arriba.

DSL es una conexión punto a punto entre el hogar y el ISP, y que por tanto todo el ancho de banda de DSL está dedicado en lugar de compartido. Una red HFC dimensiona razonablemente proporciona anchos de banda más grandes que DSL.

Los servicios siempre están activados; es decir; el usuario puede dejar su computadora encendida y éste permanecerá conectado permanentemente a un ISP mientras realiza y recibe simultáneamente llamadas telefónicas.

Puede ser mediante:

Satélite: la conexión se realiza a través de una antena parabólica que capta la señal de satélites de comunicación.

Cable: servicio prestado por las compañías que ofrecen Internet, teléfono y televisión de alta definición.

Eléctrica: utiliza la red eléctrica como línea digital de alta velocidad.

ADSL: utiliza el cable de cobre convencional dividiendo la línea en tres canales de distinta velocidad: voz, envío y recepción de datos.

Banda ancha móvil: permite el acceso a Internet sin cables. Se realiza a través de las redes de telefonía móvil utilizando dispositivos 3G y 4G.

WiMAX o LMDS: se utiliza en zonas rurales, donde el despliegue de cable o fibra sería muy costoso por la baja densidad de población.

***Clasificación de Redes***

* Diferentes clasificaciones de acuerdo a diferentes aspectos.
* Se pueden mencionar:

• Clasificación por cobertura, distancia, alcance.

• Clasificación por acceso abierto o privado.

• Clasificación por topología física.

• Clasificación por tipo de conexión/medio.

• Etc.

***Clasificación por cobertura:***

* [**Red de área local**](http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_local) (Local Área Network, **LAN**), es una red que se limita a un área especial relativamente pequeña tal como un cuarto, un solo edificio, una nave, o un avión. Las redes de área local a veces se llaman una sola red de localización. No utilizan medios o redes de interconexión públicos. Ethernet, Wi-Fi.
* [**Red de área metropolitana**](http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_metropolitana) (Metropolitana Área Network, **MAN**) es una red de alta velocidad (banda ancha) que da cobertura en un área geográfica más extensa que un campus, pero aun así limitado. Por ejemplo, una red que interconecte los edificios públicos de un municipio dentro de la localidad por medio de fibra óptica. Metro Ethernet, MPLS, Wi-Max.
* [**Red de área amplia**](http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_amplia) (Wide Área Network, **WAN**), son redes informáticas que se extienden sobre un área geográfica extensa utilizando medios como: satélites, cables interoceánicos, Internet, fibras ópticas públicas, etc. PPP, Frame-Relay, MPLS, HDLC, SONET/SDH.
* [**Red de área de almacenamiento**](http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_de_almacenamiento) (Storage Área Network, **SAN**), es una red concebida para conectar [servidores](http://es.wikipedia.org/wiki/Servidores), matrices (arrays) de discos y librerías de soporte, permitiendo el tránsito de datos sin afectar a las redes por las que acceden los usuarios. iSCSI, Fibre Channel, ESCON.
* **Red de área personal** (Personal Área Network, **PAN**) es una red de computadoras usada para la comunicación entre los dispositivos de la computadora cerca de una persona. Bluetooth, IrDA, USB.

***Clasificación: Públicas y privadas***

**Internet:** red pública global, tecnología TCP/IP.

**Intranet:** red privada que utiliza la tecnología de Internet.

**Extranet:** red privada virtualizada sobre enlaces WAN: Internet. Intranet con acceso de usuarios remotos. VPN (Virtual Private Network) IPSec, PPTP, SSL, OpenVPN. Una intranet mapeada sobre una red pública como Internet.

***Clasificación Física de Redes***

**Redes de Conmutación de Circuitos.**

**Redes de Conmutación de Tramas/Paquetes.**

• Servicios Orientados a Conexión. Circuitos Virtuales.

• Servicios NO Orientados a Conexión. Datagramas.

***Objetivos/Historia de Internet***

***Inicios de 1960’:*** Red militar para la guerra fría?? (Aún no existía TCP/IP). Packet Switching Theory: paper de Kleinrock, usar paquetes en lugar de circuitos en 1960. ARPANET: RAND Corp, Leonard Kleinrock del MIT trabajan sobre la red, BBN implementa IMPs.

***Primera vez On-Line 1969:*** Conectaba las Universidades: Stanford (SRI), Utah, UCLA, UCSB (UC Santa Barbara).

***Nuevo protocolo LAN 1973:*** Ethernet, Bob Metcalfe en Xerox PARC.

***Cambio a TCP/IP 1983:*** Desde NCP a TCP/IP. Vinton Cerf y Robert E. Kahn.

***Luego, NSFNET 1985:*** Red Cient´ıfica e Investigación, Usada por las Universidades.

***Continuando 1988:*** Comienza como negocio, nuevas oportunidades.

***Hoy 2013:*** Tele-trabajo, Redes Sociales, multimedia y todo el resto.

***Estructura de Internet***

* Estructura en Jerárquica, en Tiers.
* Capa de Acceso (Edge): Acceso Residenciales, Acceso de Organizaciones.
* Capa de núcleo (Core): dividida en diferentes niveles.

• Proveedores Regionales (Regional ISPs).

• Proveedores Nacionales.

• Proveedores Internacionales.

• Proveedores Internacionales en el Tier 1.

***Tecnologías de Internet (Redes Locales)***

* Ethernet sobre diferentes medios:

• Fibra Óptica.

• Cobre TP: Cat3, Cat5, Cat5e, Cat6a.

• Cobre Coax.

* Wireless 802.11a/b/g, Bluetooth, Infrared, Satelites.
* Tecnologías Obsoletas: Token-Ring, ARCnet, etc.

***Tecnologías de Internet (Acceso Residenciales)***

* POTs + Modems Dial-ups.
* HFC (Hybrid Fiber-Coax) Cable-modem.
* xDSL: ADSL, HDSL, SDSL.
* Wireless 802.11a/b/g.

***Tecnologías de Internet (Acceso Organizaciones)***

* Frame-Relay.
* ISDN.
* X.25.
* T1/E1, T3/E3, ...
* ATM.
* MPLS.
* Wi-MAX.
* Metro-Ethernet.

***Estructura de Internet (Core)***

* Tecnologías de Fibra Óptica, Cobre y Satélites.
* Se interconectan mediante POPs (Point Of Presence) con Proveedores.
* Entre proveedores se interconectan mediante NAPs (Network Access Point) o conexiones Directas.
* Actualmente los NAPs se los llama IXPs (Internet Exchange Point)

***Unidad 2 – Capa de Aplicación***

La Capa de Aplicación es la Capa 5 dentro de la Estructura TCP/IP. Esta capa ofrece la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP, IMAP, SMTP), gestores de bases de datos y servidor de ficheros (FTP). HTTP (para la web). DNS para la traducción de nombres de host a la direcciones IP.

Usuario -> interacción -> programas -> interactúan -> Capa de Aplicación

***Principios de los protocolos de la capa de aplicación***

Los SO, no son programas software los que se comunican, sino verdaderos procesos. Un proceso puede ser visto como un programa ejecutándose en un sistema final. Cuando lo procesos se ejecutan en el mismo sistema, se comunican entre sí utilizando comunicación interprocesos. Las reglas de comunicación interprocesos son gobernadas por el SO del sistema final. Los procesos en dos sistemas finales distintos se comunican entre sí intercambiando mensajes a través de una red de computadoras. El proceso emisor crea y envía mensajes sobre la red; el proceso receptor recibe estos mensajes y posiblemente responde enviado mensajes de vuelta. Las aplicaciones de red constan de protocolos de la capa de aplicación que definen el formato y orden en que los procesos intercambian mensajes, así como las acciones que se toman en la transmisión o recepción de un mensaje.

***Protocolos de la capa de aplicación***

Es importante diferenciar entre las aplicaciones de redes y los protocolos de la capa de aplicación. Un protocolo de la capa de aplicación es únicamente una parte de una aplicación de red. El protocolo de la capa de aplicación de la Web, HTTP (protocolo de transferencia de hipertexto [RFC 2616]), define el formato y la secuencia de los mensajes transmitidos entre el navegador y el servidor web. Correo Electrónico: servidores de correo que contienen los buzones de los usuarios; lectores de correo que permiten a los usuarios leer y creer mensajes; un estándar para la definición de la estructura de un mensaje; y protocolos de la capa de aplicación que definen cómo se pasan los mensajes entre servidores y entre servidores y lectores de correo, y como se deben interpretar ciertas partes del mensaje.

El principal protocolo de la capa de aplicación para el correo electrónico es SMTP (protocolo simple de transferencia de correo [RFC 2821]). Este protocolo es solamente una parte de la aplicación de correo electrónico.

Un protocolo de la capa de aplicación define:

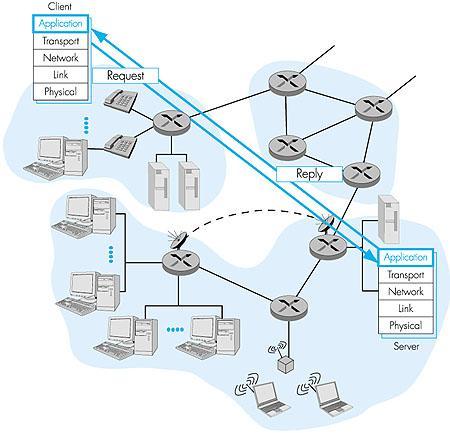
* El tipo de mensaje intercambiado (Ej. Los mensajes de petición y respuesta)
* La sintaxis de los distintos tipos de mensajes (Ej. Los campos del mensaje, y cómo estos campos están delimitados)
* La semántica de los campos (Ej. Significado de su información)
* Las reglas que determinan cuándo y cómo un proceso envía y responde a los mensajes

***Modelo Cliente/Servidor***

*Partes cliente y servidor de una aplicación*

Una aplicación de red típicamente tiene dos partes o lados: el lado del cliente y el lado del servidor. El cliente en uno de los finales se comunica con el lado del servidor en otro sistema final. Ej.: Un navegador implementa el lado del cliente HTTP, y un servidor web implementa el lado servidor de HTTP. En el correo electrónico, el servidor que envía el correo implementa el lado del cliente de SMTP, y el servidor que recibe correo implementa el lado del servidor SMTP.

Un host implementa tanto el lado cliente como el servidor de una aplicación. El host que inició la sesión es etiquetado como cliente.



**Otro modelo de comunicación:**

**P2P (Peer to Peer)**

El cual no siempre el servidor está encendido.

*Procesos que se comunican a través de una red*

Los dos procesos se comunican enviando y recibiendo mensajes. Un proceso envía mensajes a la red y recibe mensajes desde la red a través de su socket. Un Socket de proceso puede ser visto como la puerta del proceso. Cuando quiere enviar un mensaje a otro proceso en otro host, empuja el mensaje por la puerta.

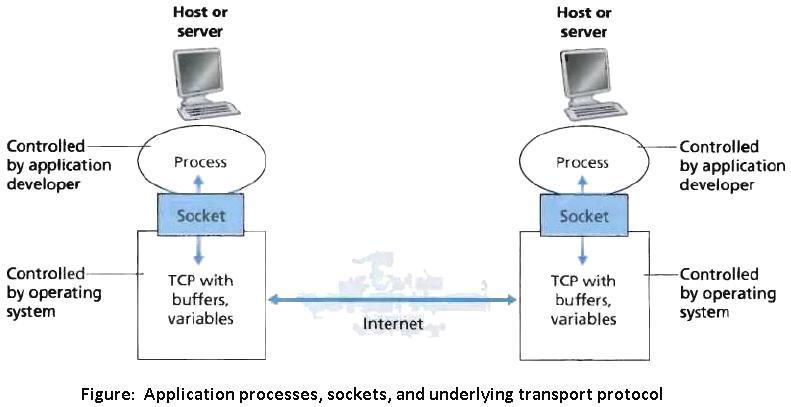
Una vez que el mensaje llega al host destino, el mensaje pasa a través de la puerta del proceso receptor, que actúa sobre el mensaje.

Un socket es una interfaz entre la capa de aplicación y la capa de transporte en un host. También se le denomina API (interfaz de programación de aplicaciones) entre la aplicación y la red, dado que el socket es la interfaz de programación con la que se construyen en internet las aplicaciones de red.

Direccionamiento de procesos

Para que un proceso de un host pueda enviar un mensaje a otro proceso de otro host, es necesario que el proceso emisor identifique el proceso receptor. Para esto, se deben especificar dos tipos de información: (1) el nombre o dirección del host, y (2) un identificador que especifique al proceso receptor en el host destino.

El host destino se identifica por la **dirección IP.** IP es una cantidad de 32bits que identifica unívocamente al host.



*Agente de usuario*

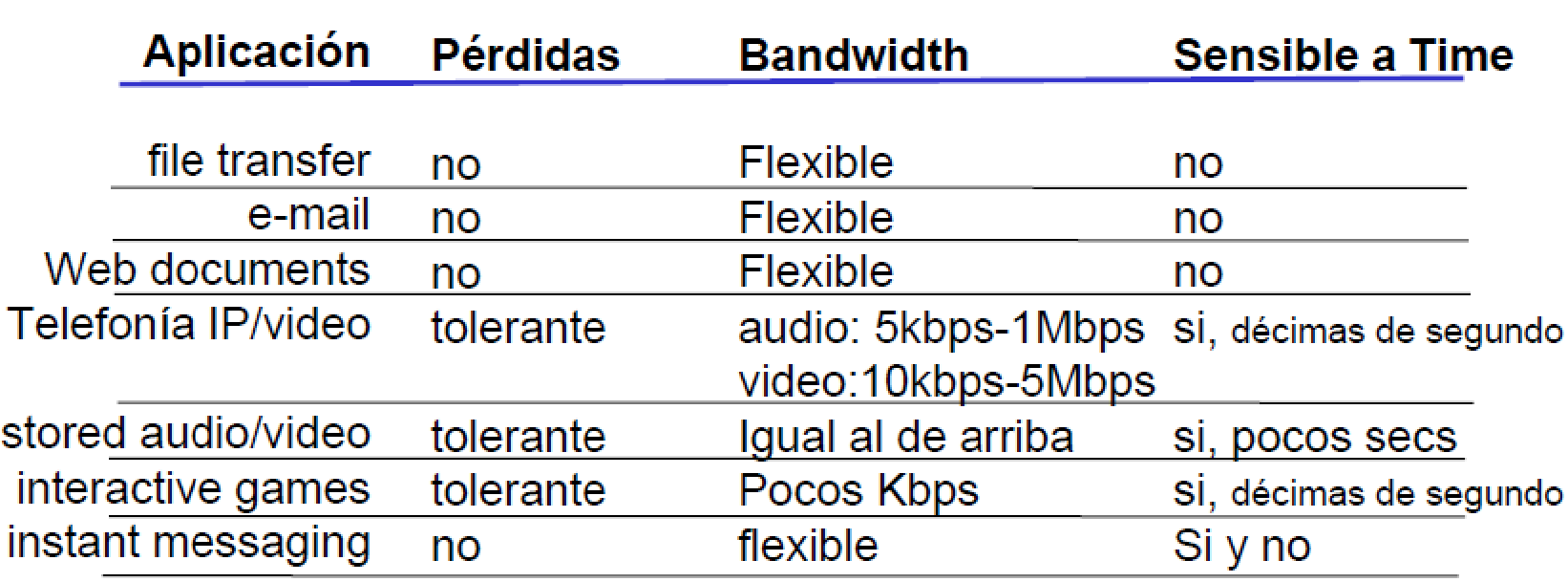
El agente de usuario es una interfaz entre el usuario y la aplicación de red. Ej. En la web, el agente de usuario es un navegador, permite al usuario ver la página web, navegar por la web, proporcionar entradas para los formularios, etc. El navegador también implementa el lado del cliente del protocolo HTTP.

En el correo electrónico el agente de usuario es un lector de correo que permite al usuario leer y componer mensajes. (Outlook). Estos proporcionan a menudo una interfaz gráfica de usuario. Estos lectores de correo a menudo utilizan 2 protocolos diferentes: SMTP para enviar correo, y un protocolo de acceso, como POP3 o IMAP, para recuperar el correo.

**Servicios que necesita una aplicación**

Se pueden clasificar los requisitos de servicio de las aplicaciones según 4 dimensiones:

* *Transferencia fiable de datos:* Algunas aplicaciones (correo, transferencias de archivos, acceso a estaciones remotas, transferencias de documentos web, etc.). Requieren una transferencia de datos completamente fiable, esto es, sin pérdida de datos. Otras aplicaciones tolerantes a pérdidas, como aplicaciones multimedia audio/video en tiempo real, pueden tolerar cierta cantidad de datos perdidos. En esas aplicaciones multimedia, la pérdida de datos puede provocar un pequeño defecto en la reproducción de audio/video, que no supone un perjuicio crucial.
* *Ancho de banda*: El correo, la transferencia de archivos, y las transferencias web son todas aplicaciones flexibles. Por supuesto, cuanto mayor sea el ancho de banda, mejor.
* *Temporización:* Las aplicaciones interactivas de tiempo real, como la telefonía, los entornos virtuales, la teleconferencia y los juegos multijugador, requieren, para ser efectivas, de estrictas restricciones de temporización sobre el envió de datos.
* *Seguridad:* Confidencialidad (cifrado), Autenticidad, e integridad.



***Servicios proporcionados por los protocolos de transporte de internet***

Las redes TCP/IP proporcionan a las aplicaciones dos protocolos de transportes, UDP (protocolo de datagrama de usuario) y TCP (protocolo de control de transmisión).

*Servicios TCP*

El modelo de servicio TCP incluye un servicio orientado a la conexión y un servicio fiable de transferencia de datos:

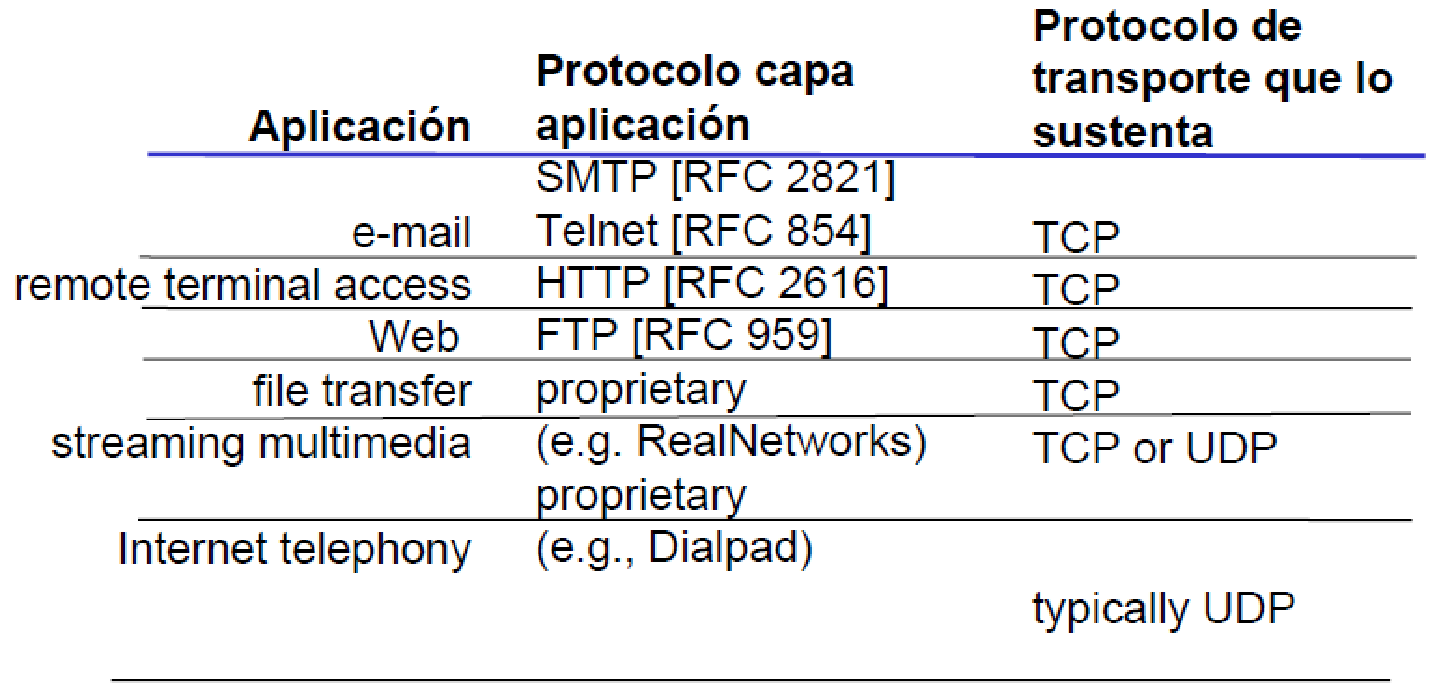
* *Servicio orientado a la conexión:* TCP hace que el cliente y servidor intercambien entre sí información de control de la capa de transporte antes de que comience el flujo de los mensajes del nivel de aplicación. Después de esta fase, se dice que existe una conexión TCP entre los sockets de los dos procesos. La conexión full-dúplex, los dos procesos pueden enviarse mensajes entre sí sobre la conexión, y al mismo tiempo. Cuando la aplicación deja enviar mensajes debe romper la conexión.
* *Servicio fiable de transporte:* se basan en TCP para enviar todos los datos sin errores y en orden apropiado.

TCP incluye un mensaje de control de congestión. Regula el proceso emisor cuando la red está congestionada entre el emisor y el receptor. Intenta limitar cada conexión TCP a su cuota justa de ando de banda de la red. TCP garantiza el envió de todos los datos. Pero no garantiza la tasa de envió ni los retardos experimentados.

*Servicio UDP*

UDP es un protocolo de transporte ligero. UDP funciona sin conexión, no hay acuerdo antes de que los dos procesos comiencen a comunicarse. Proporciona transferencia de datos no fiables; cuando un proceso envía un mensaje a un socket UDP, este NO garantiza que el mensaje llegue al proceso receptor. Los mensajes que llegan al proceso receptor, pueden hacerlo desordenado.

UDP no incluye un mecanismo de control de congestión.



Ni TCP ni UDP garantizan en la temporización.

**NOTA**

**Si dos procesos deben comunicarse:**

**a. En diferentes máquinas:** Utilizan en intercambio de mensajes a través de la red. El proceso emisor crea y envía los mensajes, y el receptor los recibe y responde si es apropiado hacerlo a través de su socket (puerta). El proceso emisor asume que existe una infraestructura de transporte al otro lado de la puerta que transportara el mensaje a través de internet hasta la puerta del proceso destino. Los protocolos de la capa de aplicación definirán el formato y orden en que deben enviarse esos mensajes, y las acciones a tomar en consecuencia a la llegada de un mensaje.

**b. En la misma máquina:** Utilizan las facilidades de comunicación inter-proceso que soporte el SO huésped (uso de señales, memoria compartida...). Las reglas de comunicación inter-proceso son gobernadas por el SO.

-------------------------------------------------------------------------------------------

**DNS (Domain Name Server) - Servicio de directorio de internet**

Los nombres de host son mnemotécnicos. Sin embargo, los nombres de host proporcionan muy poca información, si es que proporcionan alguna, sobre la localización del host dentro de internet.

Los host también son identificados por la **dirección IP**. Una dirección IP consta de de cuatro byte y tiene una rígida estructura jerárquica. Es así, porque leída de izquierda a derecha se obtiene cada vez información más específica sobre la ubicación del host en internet.

***Servicios proporcionados por DNS***

Existen dos formas de identificar un Host: por un nombre de host o por una dirección IP. Se necesita un servicio de directorio que traduzca los nombres de host a direcciones IP, esta es la principal tarea del sistema de nombres de dominio (DNS). El DNS es 1) una base de datos distribuidas implementada en una jerarquía de servidores de nombres, y 2) una aplicación de la capa de aplicación que permite que se comuniquen los host y los servidores de nombres para proporcionar el servicio de traducción. El protocolo DNS se ejecuta sobre UDP y utiliza el puerto 53.

El DNS es usado por otros protocolos de la capa de aplicación (HTTP, SMTP, FTP), para traducir los nombres de host proporcionados por los usuarios a direcciones de IP.

DNS proporciona otros servicios importantes, además de la traducción de nombres de host a direcciones IP:

* Alias del Host
* Alias del servidor de correo
* Distribución de carga

Trabaja de la siguiente manera, la aplicación invocará el lado del cliente del DNS, especificando el nombre del host que necesita ser traducido, entonces el DNS en el host del usuario entra en funcionamiento, enviando un mensaje de petición a través de la red. Después el DNS en el host del usuario recibe un mensaje DNS de respuesta que proporciona la correspondencia deseada. Esta correspondencia es entonces pasada a la aplicación, el DNS es una caja negra que proporciona un servicio sencillo y directo de traducción y consta de un gran número de servidores de nombre distribuidos por el mundo, además de constar de un protocolo de la capa de aplicación que especifica cómo se comunican los servidores de nombre y los host de peticiones.

Problemas de un diseño centralizados:

* un único punto de fallo
* volumen de trafico
* base de datos centralizada distante
* mantenimiento

El DNS es distribuido por diseño. DNS utiliza un gran número de servidores de nombres organizados de forma jerárquica y distribuida alrededor del mundo, existen tres tipos de servidores de nombres:

* **Servidores locales de nombres:** Cada ISP tiene un servidor local de nombres. Cuando un host emite un mensaje de DNS de consulta, dicho mensaje es enviado, en primer lugar, al servidor local de nombres del host. La dirección IP del servidor local de nombres se configura habitualmente manualmente. Típicamente el servidor local de nombres está cerca del cliente.
* **Servidores raíz de nombres:** cuando un servidor de nombres local no puede satisfacer una consulta de un host, el servidor local se comporta como un cliente DNS, y hace una consulta a uno de los ROOT server de nombres. Si el servidor raíz tiene un registro del nombre del host, envía un mensaje DNS de respuesta al servidor local, y entonces el servidor local envía una respuesta DNS al host peticionario. Podría no tener el registro del nombre del host, en cuyo caso conoce la dirección IP de un servidor autorizado de nombres que conoce la correspondencia para ese nombre de host particular.
* **Servidores autorizados de dominio**: Todo host se registra en un servidor autorizado de nombres. Siempre tiene un registro DNS que traduzca el nombre del host a su dirección IP.

Una consulta es una solicitud de resolución de nombres que se envía a un servidor DNS. Hay dos tipos de consultas: recursivas e iterativas.

Una ***consulta recursiva*** es aquella realizada a un servidor DNS, en la que el cliente DNS solicita al servidor DNS que proporcione una respuesta completa de la consulta. Una sugerencia de raíz es un registro de recursos DNS almacenado en un servidor DNS que indica la dirección IP para los servidores raíz DNS.

Una ***consulta iterativa*** es aquella efectuada a un servidor DNS en la que el cliente DNS solicita la mejor respuesta que el servidor DNS puede proporcionar sin buscar ayuda adicional de otros servidores DNS. El resultado de una consulta iterativa suele ser una referencia a otro servidor DNS de nivel inferior en el árbol DNS.

|  |  |
| --- | --- |
|  |  |

*Root Server:* Es un servidor de nombre de dominio y es el servidor que sabe dónde están los servidores de nombres autorizados para cada una de las zonas de más alto nivel en internet.

*Generic top-level domain (gtld):* Los dominios genéricos (.com, .name, .net, .org, .info, .biz, .aero, etc), se diferencian de los dominios territoriales (.co, .es, .tk, .tv, .fm, .ws) en que no son controlados ni gestionados por ningún país, sino por organismos gestores de internet.

*Servidor autorizado de nombres:* Es un servidor de nombres en el ISP local del host. Un servidor de nombres está autorizado para un host si siempre tiene un registro DNS que traduzca el nombre del host a su directorio IP. Cuando es consultado por un servidor raíz, el primero responde con una respuesta DNS que contiene la correspondencia al host que hizo la petición.

*Sistema de Nombres:* El espacio de nombres de Internet, sobre el cual trabaja el DNS, ha sido organizado de forma jerárquica a través de dominios, sub-dominios y nombres finales. Cada eslabón de la jerarquía es un dominio o sub-dominio (también llamados ZONAS) que contiene un espacio de nombres bajo esa denominación. Existen dominios de primer nivel en la jerarquía, conocidos como TLD (Top Level Domains), que están predefinidos. Actualmente, los TLDs son manejados por el IANA (Internet Assigned Numbers Authority) a través del ICANN (Internet Corporation for Assigned Names and Numbers).

El registro de los nombres no está concentrado todo en el ICANN. Este se encarga de administrar los TLD, y luego delega registros por cada país. Por ejemplo, para la Argentina la entidad encargada es NIC.AR, que depende de la Cancillería de la Nación. Es posible delegar dominios raíces a otras organizaciones. El ICANN también se encarga de administrar las direcciones IPv4, direcciones IPv6 y los números de sistemas autónomos (AS). Estas tareas no se concentran todas en el ICANN, sino que son delegadas a los Registros de Internet Regionales, en ingles RIRs (Regional Internet Registry). Estas son organizaciones que tienen a cargo diferentes regiones del mundo.

*Los TLD (Top Level Domains) se podrían clasificar en 3 grupos:*

**GTLD (Generic TLD),** contienen dominios con propósitos particulares, de acuerdo a diferentes actividades. Para 1980, los gTLD eran: .com, .edu, .gov, .int, .mil, .net y .org, pero solo .com, .net y .org tenían libre el registro, el resto estaban dedicados, únicamente, a organizaciones de los Estados Unidos u organizaciones Internacionales. Debido a esto, se creía que todos los registros (incluso los 3 libres) se encontraban en Estados Unidos. Las políticas de uso y registro pueden estar definidas por el ICANN, en ese caso se llaman Unsponsored TLD, o, si son definidas por otra organización a la cual se le delega el control se los llama Sponsored TLD. Para 2008, según [gTLD-ICANN], los gTLD son los mostrados en la figura 2.

**ccTLD (Country-Code TLD)**, contienen dominios delegados a los diferentes países del mundo. Los códigos de los países están codificados en dos símbolos, habitualmente letras, y actualmente se admiten símbolos o letras de otros alfabetos. Se encuentra por ejemplo .ar (Argentina), .tv (Tuvalu), .zw (Zimbabwe), .uk (Reino Unido), .ru (Rusia), .us (Estados Unidos), etc. En la figura 3 se muestran algunos ejemplos [ccTLD-ICANN].

**.ARPA TLD**, es un dominio especial, usado internamente por los protocolos, creado para resolución de reversos: de direcciones a nombres. Se muestra integrado en la figura 3 como “in-addr.arpa”.

***Delegación de Sub-dominios/Zonas:***

Una vez definido el espacio de nombres, y su administración, es necesario asignarlo a diferentes servidores de manera distribuida. Para 1993, había 8 servidores, llamados ROOT Servers (Servidores Raíces), encargados de atender la raíz del servicio. Estos eran administrados por InterNIC y se encargaban de delegar cada una de las zonas generadas para los TLD, tanto gTLD como ccTLD. La delegación consiste en saber las direcciones IP de los servidores que se encargan de resolver (o sub-delegar) las zonas de manera autoritativa. Un servidor autoritativo tiene toda la información para una zona, puede producir cambios sobre la misma y es el que tiene la última versión.

En la actualidad, existen 13 ROOT Servers distribuidos en todo el mundo, 7 de los cuales no son servidores únicos, sino que trabajan con redundancia y las replicas están distribuidos geográficamente. Con esta redundancia, combinada con Ruteo Anycast, se logra que los requerimientos se atiendan según la proximidad, logrando mejores tiempos de respuesta. La cantidad de servidores raíces está acotada a 13 debido a la limitación de 512 bytes de un mensaje de DNS sobre UDP.

**Resolver**: Toda computadora con un SO utiliza un Resolver (Resolución de nombres). Este se encarga de encontrar la resolución de nombres para todos los programas que solicitan una resolución. Esta información es precisamente la traducción de un nombre de Internet en su dirección de IP equivalente, u otra información para los DNS.

El funcionamiento del DNS se realiza en un esquema cliente/servidor, siendo el cliente cualquier aplicación que requiera la resolución de nombres. El código (instrucciones de maquina) del cliente se lo agrupa en un módulo llamado Resolver. El Resolver se lo podría considerar como un agente encargado de resolver los nombres a solicitud del cliente, por ejemplo: un servidor web, el cliente telnet, un navegador web, un servidor de mail, etc. Este agente, generalmente, no se implementa como un servicio activo, sino como un conjunto de rutinas encapsuladas en una biblioteca de funciones que se enlaza y usa conjuntamente con la aplicación.

Este tipo de resolver no realiza ninguna forma de caching y se lo suele llamar Stub Resolver. El encargado del caching, en este caso, es el Servidor Local. Existen otras implementaciones que tienen un resolver activo, Smart Resolver, funcionando en cada equipo como si fuese un Servidor Local, que permiten realizar caching u ofrecer funcionalidades extras.

***Tipos de registros de DNS:***

***a. PTR:*** IP → nombre de dominio. Es inversa del A. Traduce IPS en nombres de dominio.

***b. A:*** Nombre de dominio → IP. Traduce nombres de dominios.

***c. NS:***  Name Server: define la asociación que existe entre un nombre de dominio y los servidores de nombres. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.

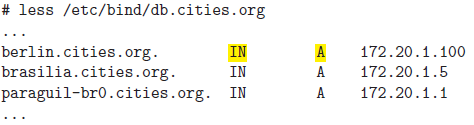
***d. MX:***  Mail Exchange (Intercambiador de mail): define el lugar donde se almacena el correo que recibe en el dominio.

***e. SOA:*** Comienzo de autoridad: da información de una zona de/los dominios

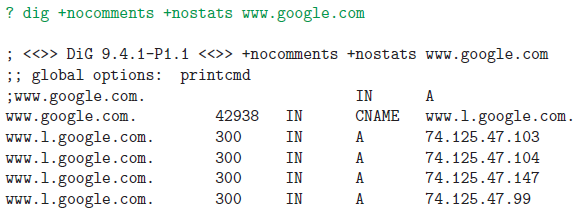
***f. CNAME:*** Nombre canónico: se usa para crear nombres de hosts adicionales, o alias, para los host de un nuevo dominio.

Un servidor de DNS almacena la información con la que trabaja en una base de datos (DB) local. En la mayoría de los casos no es una base de datos relacional, aunque podría usarse. La DB es una estructura propia del sistema que se genera en memoria una vez que arranca el servicio y se configura desde los archivos que residen en el sistema de archivos local. Dentro de la base de datos, la información se organiza en registros llamados Resource Record (RR). Cada uno de estos registros puede guardar diferente tipo de información. En la clasificación de RR, antes de los Tipos existen Clases. Es de nuestro interés trabajar, únicamente, con la clase INTERNET (IN). Las clases permiten utilizar el protocolo de DNS no solo para IP. Ejemplos de estos tipos se describen a continuación:

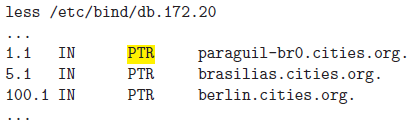
**Registros A** (Address), son registros que mapean de un nombre de dominio a una dirección IP. Son los más comunes.



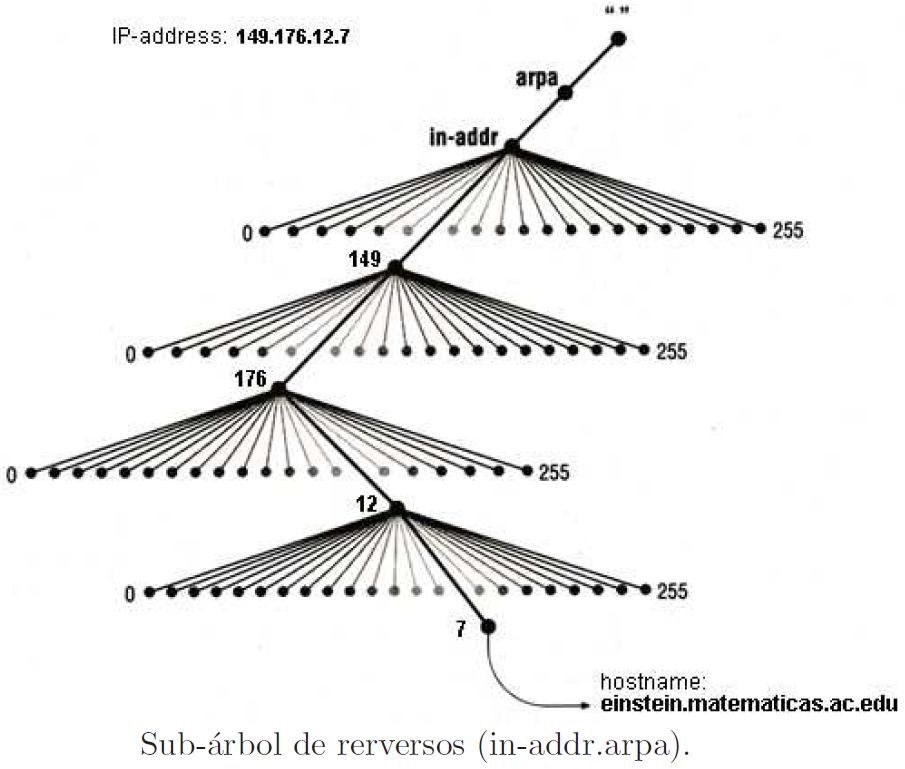
Pueden existir varios registros (A) con el mismo nombre, por ejemplo, para realizar balance de carga de un servidor muy accedido. Para aprovechar esto, el servidor de DNS debería responder con una lista de direcciones IP, siempre alternando el resultado con algún criterio, por ejemplo de forma circular. A continuación, usando la herramienta dig(1), se muestra un ejemplo de balance para los servidores de Google.

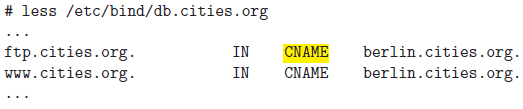


**Registros PTR** (Pointer), estos registros mapean direcciones IP a nombres de dominio. Son el inverso de los registros (A), por esto, se los suele llamar reversos. Trabajan en el dominio especial in-addr.arpa. Por ejemplo:

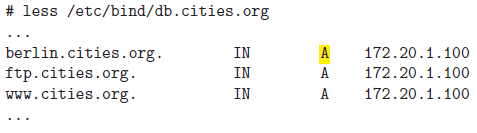


Los registros (PTR) deben estar en un subárbol (dominio) separado que, como se mencionó más arriba, se llama in-addr.arpa. Esto se debe a que la búsqueda se realiza usando la dirección IP y no el nombre. Aunque, la información existe en los registro directos (A), no se puede generar un mecanismo de búsqueda organizado, ya que dada una dirección IP no hay forma de saber en dónde, que zona, esta asignada. Para este propósito, está el sub-árbol in-addr.arpa que organiza las direcciones por octeto de las direcciones IP, generando un árbol con la estructura mostrada en la figura 9. De esta manera, se provee un mecanismo a modo de índice de búsqueda.

  
**Registros CNAME** (Canonical Name), son registros que mapean de un nombre de dominio a otros nombres. Se los conoce como aliases, debido a que dado un nombre indican el nombre original. Esta funcionalidad podría, también, resolverse con varios registros (A). El nombre, al que apunta, debería ser el nombre original, conocido como canónico.

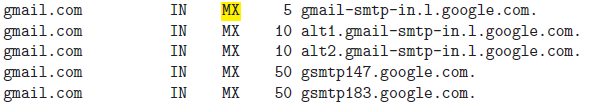


Si no se usasen registros (CNAME) se podría generar la siguiente configuración:



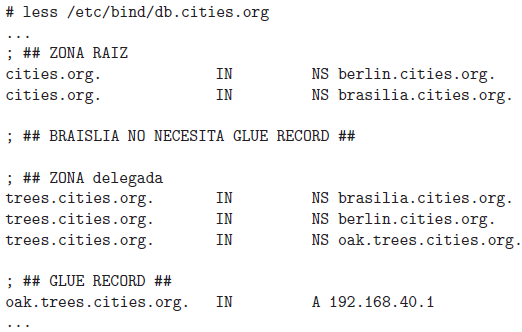
**Registros MX** (Mail Exchanger), son registros que indican, para un nombre de domino, cuales son los servidores de mail SMTP encargados de recibir los mensajes para ese dominio. De esta forma, no es necesario especificar el servidor completo de mail donde se encuentra la casilla destino, alcanza con indicar, solamente, el dominio. El servidor de mail SMTP que envía el mensaje deberá consultar, vía el servicio de DNS, cuales son los servidores SMTP receptores para el dominio dado. Se puede detallar una lista de servidores y asignarles prioridades. Los valores más bajos son las mejores prioridades. Así, al momento de hacerse el envío del mensaje, si el servidor primario no está activo se podría recurrir a otro de menor prioridad para enviar el e-mail. Con la asignación de prioridades con igual valor se podría lograr un balance de carga entre los distintos servidores SMTP.

Suponiendo que se desea enviar un e-mail a la cuenta <pepe@gmail.com>, los servidores SMTP destinos en orden, de acuerdo a su prioridad, serían los siguientes:



**Registros NS** (Name Server), los registros (NS) indican los servidores de nombre autoritativos para una zona o sub-dominio. A partir de esto, se puede lograr una delegación de sub-dominios. A diferencia que los registros (MX), estos no llevan asociado una prioridad, todos los servidores tienen la misma precedencia. Para lograr un mejor balance, al ir respondiendo se debería ir rotando el orden con el que se entregan los servidores autoritativos para una zona. Al tener varios servidores para un mismo dominio no es necesario configurar a todos con los mismos datos. El software de DNS permite asignar roles de Servidor Primario y Servidor(es) Secundario(s). De esta forma, solo se requiere configurar el primario y luego el secundario obtendrá una copia de la base de datos del servidor maestro/primario. Es importante que los servidores estén actualizados, por eso debe existir algún mecanismo para mantenerlos sincronizados. El Servidor Primario debería avisar a todos los Servidores Secundarios cuando se realiza un cambio, así estos re-copian la base de datos de nombres desde el servidor maestro. La comunicación entre servidores se realiza vía el mismo protocolo de DNS, con la diferencia que se hace sobre TCP en lugar de UDP, esto se debe a que, generalmente, los datos transmitidos ocupan más de 512 bytes (máximo para DNS sobre UDP). Esta operación se llama Transferencia de Zona o, en ingles, Zone Transfer.

Se recomienda sobre una configuración de un servidor de DNS, en el cual se delegan las zonas, y las direcciones IP de los servidores delegados están en la zona delegada, agregar para cada registro (NS) un registro (A), indicando la dirección IP del servidor delegado. Esto se utiliza para agilizar las búsquedas y ahorrar una doble consulta. Estas entradas (A) se las conoce como GLUE RECORDS, ya que “pegan” un nombre de un servidor de nombres con su correspondiente dirección IP. En el ejemplo de cities.org se podría tener dos zonas: cities.org y trees.cities.org.



Por lo que contiene el hostname (dominio), y el hostname de un name server (servidor autorizado) a quien seguir consultando. Permite continuar las consultas hasta obtener la traducción definitiva.

**Registros SOA** (Start Of Authority), los registros (SOA) se crean por cada zona o sub-zona que brinda el servicio de DNS. En este registro se especifican los parámetros globales para todos los registros del dominio o zona. Solo se admite un registro (SOA) por zona.

**¿Qué función cumple en Linux/Unix el archivo /etc/hosts o en Windows el archivo \WINDOWS\system32\drivers\etc\hosts?**

Es un vestigio de los tiempos que solo había unos pocos dominios y se enviaba la lista con todos ellos y sus respectivos IPS en un archivo llamado host.

En la actualidad todas las peticiones se realizaban a los servidores DNS, no obstante los sistemas actuales mantienen este archivo, de modo que es posible modificar a mano y que IP deben resolver cada dominio.

**[EXTRA]**

Organización del DNS, es un sistema distribuido y jerárquico, organización mediante dominios, sub-dominios y host o servicios, el funcionamiento lo controla la IANA a través del ICANN (Internet Corporation for Assigned Names and Numbers). Existen organizaciones paralelas: Open Root Server Network (ORSN), OpenNIC.

Mientras que para la asignación de direcciones IP no se delega a los países, la delegación de nombres ocurre mediante RIRs (Regional Internet Registers):

* American Registry for Internet Numbers (ARIN).
* RIPE NCC -Europa y parte de Asia- (RIPE).
* Asia-Pacific Network Information Centre (APNIC).
* Latin American and Caribbean NIC (LACNIC).
* African Network Information Centre (AfriNIC).

Ejemplo de delegación de autoridad (ada.info.unlp.edu.ar):

* “Ada” fue registrada por la administración de la red de la Facultad de Informática.
* El administrador de la Facultad obtuvo previamente la autoridad sobre el dominio “info.unlp.edu.ar”. a partir de la administración de la universidad UNLP.
* La Universidad obtuvo autoridad sobre el dominio “unlp.edu.ar” a partir de la administración de “edu.ar”, RIU (Red Inter-universitaria).
* La RIU obtuvo autoridad sobre “edu.ar” a partir de la delegación de la Cancillería o el ente a cargo de “.AR” (Argentina).
* La administración de nombres en la Argentina, sea la Secretaría Legal y Técnica u otro ente obtuvo la autoridad delegada a partir del IANA o ICANN.

***Aplicaciones de red***

La WEB, su protocolo de la capa de aplicación HTTP. El sistema de nombres de dominio (DNS), proporciona un servicio de directorio para Internet. La mayoría de los usuarios invocan un DNS indirectamente por medios de otras aplicaciones. EL DNS muestra claramente cómo se puede implementar en internet una base de datos distribuida.

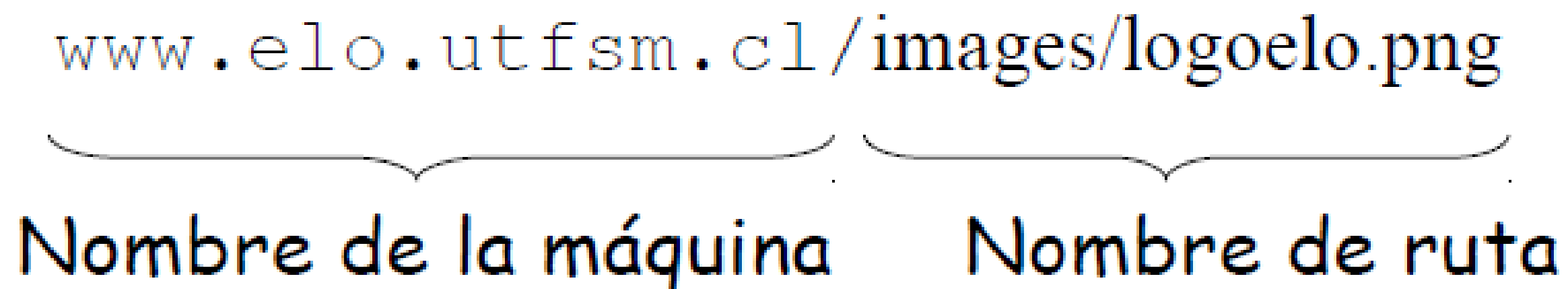
***La Web y HTTP***

***Introducción a HTTP***

El protocolo de transferencia de hipertexto (HTTP), está en el corazón de la web. [RFC 1954] y [RFC 2616]. HTTP está implementado en dos programas: un programa cliente y otro servidor. Los programas clientes y servidor, que se ejecutan en sistemas finales diferentes, conservan entre sí intercambiando mensajes HTTP. HTTP define la estructura de estos mensajes y cómo se realiza el intercambio entre el cliente y el servidor.

Página web (documento), consta de objetos. Un objeto es simplemente un archivo. (Ej. Un archivo HTML, una imagen, etc) que es direccionable por un único archivo url la mayoría de las páginas web están formadas por un archivo HTML base y diversos objetos diferenciados. Cada url tiene dos componentes: el nombre del host del servidor que alberga el objeto y el nombre de la ruta del objeto.

Ejemplo:



Tiene el nombre del host (nombre de maquina) y el nombre de ruta. Un navegador es un agente de usuario para la web. Los navegadores web también implementan el lado cliente HTTP.

Un servidor web alberga objetos web, cada uno de ellos direccionables por una url. También implementan el lado servidor de HTTP.

HTTP define como los clientes web demandan páginas web, y cómo los servidores transfieren estas páginas web a los clientes. Cuando un usuario pide una página web, el navegador envía al servidor un mensaje HTTP de petición de los objetos de la página. El servidor recibe la petición y responde con un mensaje HTTP de respuesta que contiene los objetos. Un servidor web que ejecuta HTTP 1.1 puede hablar con un navegador que ejecuta la 1.0 y un navegador con la 1.1 puede hablar con un servidor con la 1.0. Http utiliza TCP, inicia una conexión TCP con el servidor. Una vez establecida la conexión, los procesos del navegador y del servidor acceden a TCP por medio de sus interfaces de socket.

***Conexiones no persistentes y persistentes***

HTTP puede utilizar tanto conexiones no persistente como persistentes. HTTP/1.0 utiliza conexiones no persistentes. No persistentes sólo se puede transferir un único objeto web sobre una conexión TCP. HTTP/1.1 utiliza conexiones persistentes pueden ser configurados para utilizar conexiones no persistentes.

***Conexiones No persistentes***

Supongamos que el usuario ingresa URL: www.someSchool.edu/someDepartment/home.index

(Contiene texto, referencias a 10 imágenes jpeg)

**1a.** Cliente HTTP inicia una conexión TCP al servidor HTTP (proceso) en www.someSchool.edu en puerto 80

**1b.** Servidor HTTP en host www.someSchool.edu esperando por conexiones TCP en puerto 80“acepta conexión, notifica la cliente

**2.** Cliente HTTP envía mensaje de requerimiento (conteniendo el URL) por el socket de la conexión TCP. EL mensaje indica que el cliente quiere el objeto some Department/home.index

**3.** El servidor HTTP recibe el mensaje de requerimiento, forma el mensaje de respuesta que contiene el objeto requerido y envía el mensaje por su socket.

**4.** Servidor HTTP cierra la conexión.

**5.** Cliente HTTP recibe el mensaje respuesta que contiene el archivo html y despliega el html. Analizando el archivo html file, encuentra 10 referencias a objetos jpeg

**6.** Pasos 1-5 son repetidos para cada uno de los 10 objetos jpeg.

HTTP solo definen el protocolo de comunicación entre los programas HTTP cliente servidor. Cada una de las conexiones TCP es creada después de que el servidor envíe el objeto. Cada conexión TCP transporta exactamente un mensaje de petición y uno de respuesta. Por la que se generan 11 conexiones TCP.

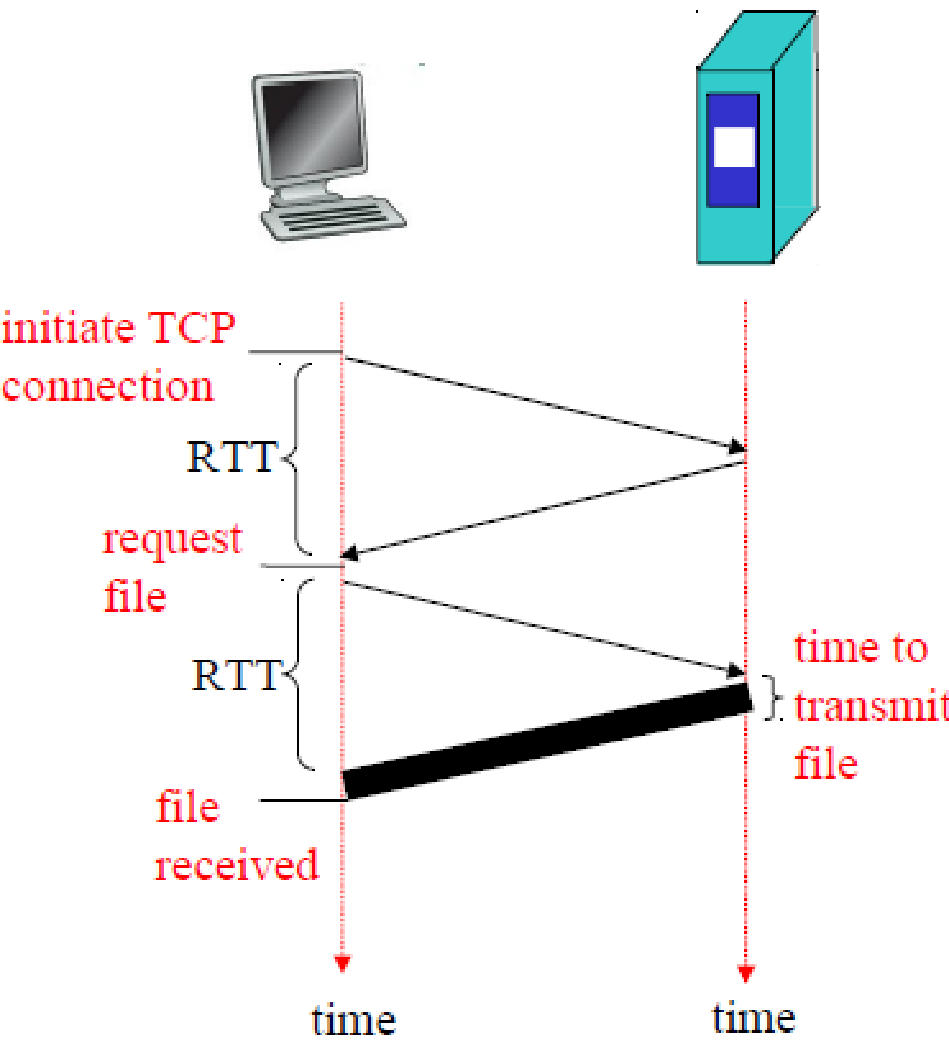
**Definición de RTT:** tiempo ocupado en enviar un paquete pequeño desde el cliente al servidor y su regreso.

**Tiempo de respuesta:**

- Un RTT para iniciar la conexión

- Un RTT por requerimiento HTTP y primeros bytes de la respuesta

- Tiempo de transmisión del archivo total = 2RTT+tiempo de transmisión



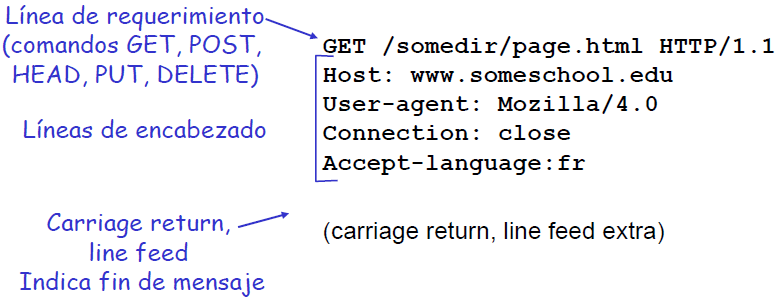
***Conexiones persistentes***

El servidor deja abierta la conexión TCP después de enviar una respuesta. Una página web completa puede ser enviada sobre una única conexión Tcp persistente. Múltiples páginas web que residen en el mismo servidor pueden ser enviadas desde el servidor hasta un mismo cliente sobre una única conexión TcP persistente.

***Relacione DNS con HTTP. ¿Se puede navegar si no hay servicio de DNS?***

En el caso más simple (GET), un navegador contacta a un servidor web directamente para obtener una página. El navegador extrae el hostname de la URL, usa DNS para mapear el nombre en una dirección IP y usa la dirección IP para establecer una conexión TCP al servidor. Una vez que la conexión está establecida el navegador y el web server usan HTTP para comunicarse; el navegador manda un pedido (HTTP REQUEST) para recibir una página determinada y el servidor responde (HTTP RESPONSE) con una copia de la página.

HTTP Request, en un mensaje de solicitud, el navegador manda un comando HTTP como pedido de una página web al servidor. Existen diversos tipos de comandos de solicitud como GET, POST, HEAD, PUT, DELETE, y otras extensiones que se puedan definir sin modificar el protocolo aunque no se puede suponer que el receptor las reconozca. El pedido consiste en una línea de texto que empieza con el tipo de solicitud, seguida de la URL, la versión de HTTP utilizada y el CRLF:



Un mensaje de requerimiento completo comienza con una “Request-line” que tiene el tipo de solicitud, uno o más “Header-lines” que contienen campos con sus respectivos valores y un “Entity body” o cuerpo (opcional):

Request-Line= Method SP1 URL SP HTTP-Version CRLF2

Header-field-name: value CRLF

…

Header-field-name: value CRLF

CRLF

Entity body

En caso de que haya que transportar valores de entrada, como por ejemplo información perteneciente a un formulario de la página, la forma de enviarlos en el requerimiento pueden ser 2. La primera y más segura, es transportar los datos en el cuerpo o Entity body del mensaje y la otra, menos segura, es colocar la entrada en campos dentro de la URL de la línea de requerimiento o Request Line.

Se podría navegar si no contamos con un DNS, pero esto condicionaría a que la URL contenga la dirección IP del servidor al que quiere acceder (y el puerto en caso de ser necesario).

HTTP Response, un mensaje de respuesta completo consta de una línea de estado “Status-Line” seguida por una o más cabeceras “Header-lines” de datos generales con sus respectivos valores, seguidas por un cuerpo de entidad “Entity body” (opcional). Un mensaje de respuesta completo comienza con una Status-Line que tiene el formato descrito a continuación y un cuerpo con datos relacionados al contenido que transporta:

Status-Line= HTTP-Version SP Status-Code SP Reason-Phrase CRLF

Header-field-name: value CRLF

…

Header-field-name: value CRLF

CRLF

Entity body

El valor de “HTTP-Versión” es el número de versión HTTP utilizado por el emisor, “Status-Code” es un entero de 3 dígitos que indica la respuesta a una solicitud recibida y “Reason-Phrase” proporciona una breve explicación del código recibido. En la siguiente tabla se organizan los códigos en diferentes categorías:

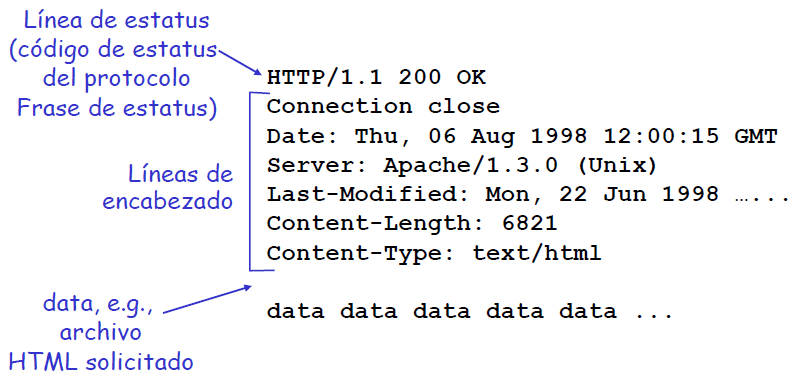
El código de estado es un entero de 3 dígitos:

* 1xx: Informativos (no usados actualmente)
* 2xx: Éxito
* 3xx: Redirección
* 4xx: Error de cliente
* 5xx: Error de servidor

Los más comunes:

* 200 OK, solicitud exitosa, la respuesta se envía en el cuerpo.
* 301 Moved Permanently, se movió el objeto requerido, la nueva ubicación es especificada luego en el mensaje en el campo “Location”
* 303 See Other, el recurso se ha movido a otra URL (Dada en el header Location).
* 400 Bad request, requerimiento no entendido por el servidor.
* 404 Not Found, el recurso no se encontró en el servidor.
* 500 Server Error, error no esperado en el servidor.
* 503 Service temporarily unavailable, el servidor se encuentra caído.
* 505 HTTP versión not supported, versión del formato no soportada por el servidor.

Ejemplo de HTTP Response:



Los métodos de solicitud indican las acciones que se desea efectuar sobre el recurso identificado. Los más comunes que define HTTP son:

* **GET**, pide una representación del recurso especificado. Por seguridad no debería ser usado cuando los datos son sensibles, ya que transmite información agregando parámetros a la URL.
* **POST**, similar al GET, pero además, envía los datos en el cuerpo del mensaje para que sean procesados por el recurso identificado.
* **HEAD**, pide una respuesta idéntica a la que correspondería a una petición GET, pero sin el cuerpo de la respuesta. Esto es útil para la recuperación de información escrita en los encabezados de respuesta, sin tener que transportar todo el contenido.
* **PUT**, sube, carga o realiza un upload de un recurso especificado (archivo), es el camino más eficiente para subir archivos a un servidor. La desventaja del método PUT es que los servidores de hosting compartido no lo tienen habilitado.
* **DELETE**, borra el recurso especificado.

***Interacción usuario-servidor: autorización y cookies***

A menudo es deseable que un sitio web identifique a los usuarios, bien porque el servidor desee restringir el acceso de los usuarios, o porque quiera servir cierto contenido en función de la identidad de usuario. HTTP proporciona dos mecanismos que ayudan al servidor a identificar a los usuarios: autorización y cookies.

***Autorización:***

Muchos sitios obligan a los usuarios a proporcionar un nombre de usuario y una palabra clave para acceder a los documentos del servidor. La petición y recepción de las autorizaciones se realiza habitualmente por cabeceras HTTP y código de estatus especiales.

*Cookies:*

Esta tecnología tiene cuatro componentes: 1) una línea de cabecera de cookie en el mensaje HTTP de respuesta; 2) una línea de cabecera de cookie en el mensaje de HTTP de petición; 3) un archivo de cookie que se almacena en el sistema del usuario y que es gestionado por el navegador del usuario; 4) una base de datos de respaldo en el sitio web.

Las cookies pueden ser utilizadas para autenticar usuarios.

Son altamente controvertidas, dado que pueden ser vistas como una infracción de la privacidad de los usuarios. Una web puede saber mucho sobre un usuario por medio de una combinación de cookies e información suministrada por el usuario, pudiendo potencialmente vender la información a terceros.

***COOKIES,*** como HTTP es un protocolo sin estados (no almacena el estado de la sesión entre peticiones sucesivas), existen las cookies que pueden usarse para asociar estado a la sesión, proporcionando una manera de conservar cierta información entre peticiones del cliente. Es información que el navegador guarda en memoria o en el disco duro dentro de ficheros texto, a solicitud del servidor. El hecho de ser almacenadas en el cliente, libera al servidor de una importante carga, haciendo que el cliente devuelva la información al servidor en siguientes peticiones. Incluyen datos generados por el servidor, o datos introducidos en un formulario por el usuario, enviados al servidor y reenviados por éste al cliente. Por lo que el fin es almacenar información importante para el servidor que le permite almacenar y recuperar información acerca de sus visitantes. Sus principales tipos y funciones son:

* Llevar el control de usuarios (Cookie de sesión); cuando un usuario introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciéndolas para cada página del servidor. Sin embargo, una cookie no identifica a una persona, sino a una combinación de computador-navegador-usuario.
* Almacenar información sobre los hábitos de navegación del usuario o preferencias (Cookies persistentes). Esto puede causar problemas de [privacidad](https://es.wikipedia.org/wiki/Privacidad) y es una de las razones por la que las cookies tienen sus detractores, ya que permiten que el sitio aprenda mucho sobre uno. Sus posibles usos son:
  + Comercio electrónico (carrito de compras).
  + Personalización de páginas (idiomas).
  + Las compañías de avisos obtienen información a partir de los datos que almacenan los sitios web.

Cuando el servidor necesita enviar una cookie, la cabecera HTTP del mensaje respuesta seria “Set-Cookie” y respetaría el siguiente formato:

**“Set-Cookie:”**

**“Nombre=valor;” -> Nombre de la cookie y valor**

**“Expires=fecha;” -> Fecha de caducidad**

**“Path=camino;” -> Camino**

**“Domain=dominio;” -> Dominio**

**“Secure” -> sólo se transmite sobre canales seguros (HTTPS).**

Las información incluida indica el Nombre y Valor de la cookie, Fecha de caducidad donde el navegador conservará y recuperará la cookie sólo si su fecha de caducidad aún no ha expirado (si no se especifica, caduca cuando el usuario salga de la sesión y es eliminada), Camino de las aplicaciones con acceso a la cookie (si no se especifica, toma como camino el directorio de la aplicación que la originó), Dominio (completo o parcial) de los servidores con acceso a la cookie (si no se especifica ningún dominio, el navegador sólo devolverá la cookie a la máquina que la originó), y atributo Secure indicando que la cookie sólo será transmitida a través de un canal seguro, con SSL. Ejemplo:

Set-Cookie: unnombre=unvalor; expires=Mon, 30-Jan-2001 12:35:23 GMT; path=/dir; domain=mi.dominio.com; secure

Cuando un cliente envía una cookie, la cabecera HTTP del mensaje seria “Cookie” y respetaría el siguiente formato:

“Cookie:”

“Nombre1=valor1;” -> Nombre de la cookie y valor

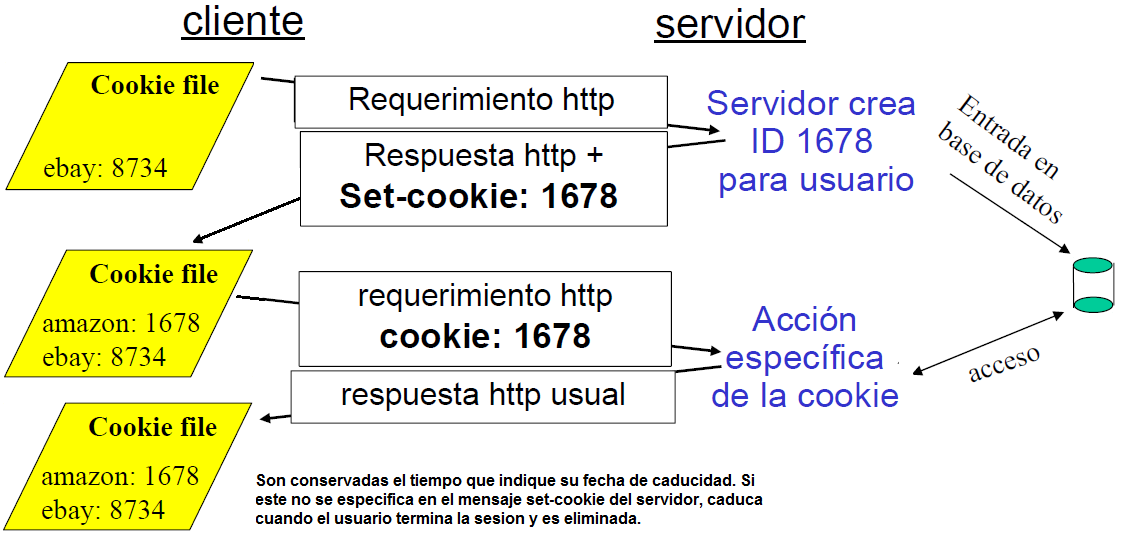
“Nombre2=valor2;” -> Nombre de la cookie y valor

…

Cuando un cliente solicita una URL, buscará en su lista de cookies aquellas que coincidan con ese dominio y con ese camino. Dentro de la cabecera “Cookie”, las cookies se ordenan de más a menos específicas (según camino) siendo estas enviadas para su posible uso en el servidor. Las cookies caducadas no son consideradas. Ejemplo:

Cookie: unnombre1=unvalor1; unnombre2=unvalor2; …

Proceso de uso de Cookies:

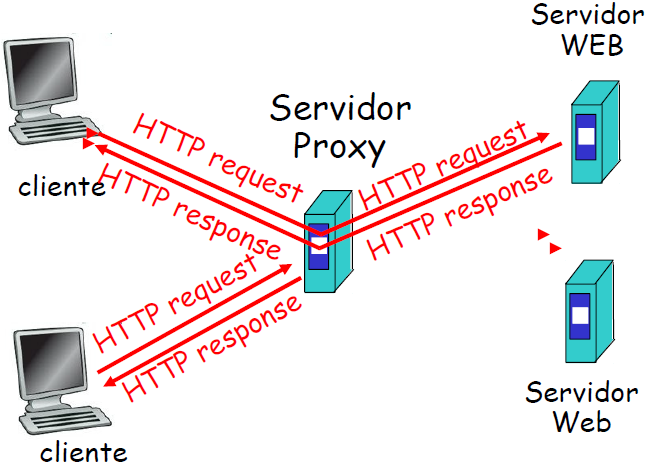


Por su diseño, las cookies tienen las siguientes limitaciones:

* Máximo de trescientas cookies en el disco, si llega el número 301, se borra la más antigua.
* Tamaño máximo de 4 Kbytes por cookie (nombre y valor).
* Veinte cookies máximo por servidor o dominio.
* Ninguna máquina que no encaje en el dominio de la cookie podrá acceder a ella.

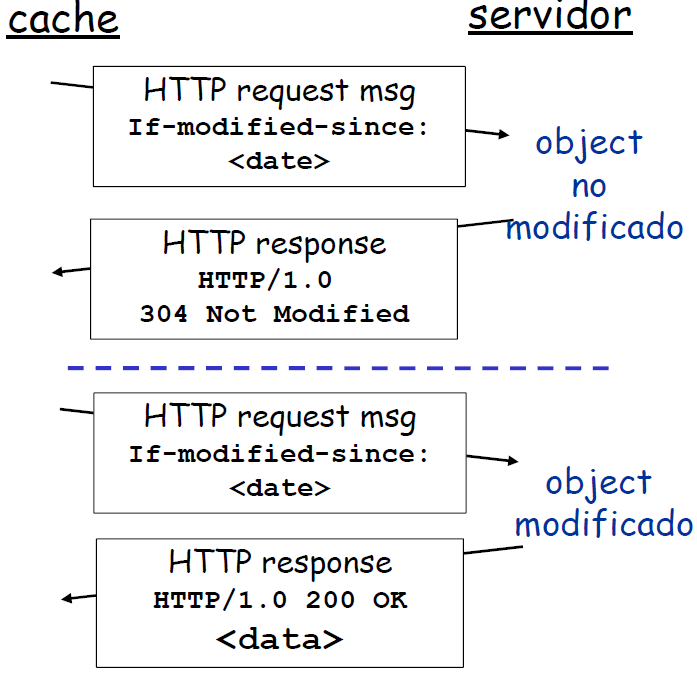
***WEB CACHING O SERVIDORES PROXY***

Tienen como objetivo satisfacer el requerimiento del cliente sin involucrar al servidor destino. El servidor cache actúa como cliente y como servidor. Típicamente el cache está instalado por el ISP4 que puede ser una universidad, compañía, un ISP residencial o de cualquier otra institución. Usuario configura el browser para tener un acceso web vía cache, el browser envía todos los requerimientos HTTP al cache, si el objeto esta en cache, este retorna el objeto; sino el cache requiere los objetos al servidor web, una vez obtenido, lo almacena y retorna al cliente.



El uso de esta técnica reduce el tiempo de respuesta de las peticiones al cliente, así como también reduce el tráfico en la red de la institución por lo que permite a proveedores de contenidos con escasos recursos de infraestructura (debido a su elevado costo) entregar contenido de forma efectiva.

Esta técnica es llevada a cabo por medio de GET condicionales entre en web cache y el proveedor de servicios. Ante un requerimiento por parte del cliente al web cache, éste especifica al proveedor de servicios la fecha de la copia en un requerimiento HTTP condicional “if-modified-since: <date> ”. Si la copia del web cache es la última, el servidor responde un código “304 Not Modified”, en caso contrario entrega la última versión del objeto. Por lo que todo este proceso evita que el servidor tenga que enviar un objeto si el web cache tiene la versión actualizada.



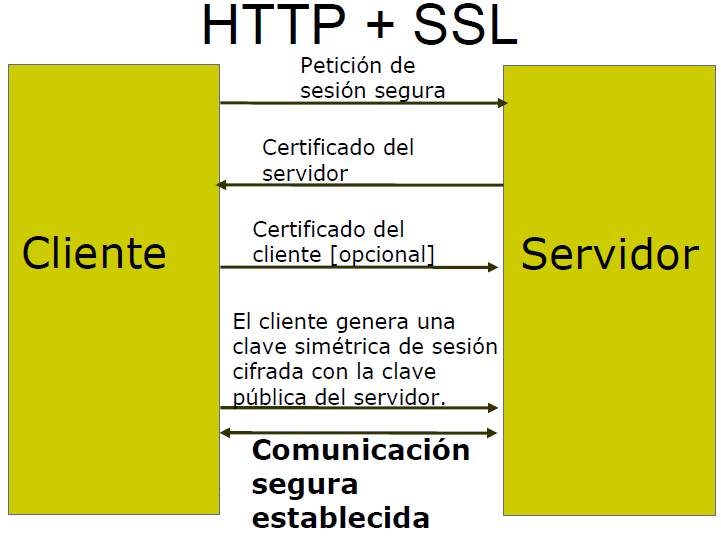
***HTTPS***

Es un protocolo de transferencia de recursos web de manera segura a través de la unión de los protocolos HTTP y SSL (Secure Socket Layers). Mediante este último protocolo se genera un canal encriptado para dicha aplicación usando el puerto de transporte TCP 443. Es usado principalmente para:

* Transacciones comerciales.
* Operaciones bancarias.
* Transmisión de datos de manera privada.
* Administración remota.
* Webmail.

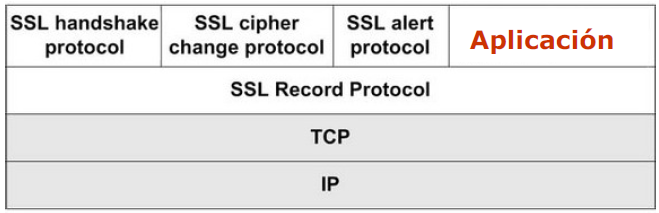
Estos servicios requieren un marco de seguridad en la comunicación ya que implican movimientos de información crítica. El objetivo de este canal encriptado es lograr:

1. **Privacidad:** Garantizar que la información sólo pueda ser leída por los miembros de la conversación.
2. **Autenticación:** Evitar que se falsifiquen los mensajes por algún impostor (ataques man in the middle). El servidor siempre debe autenticarse mediante un despliegue de infraestructura de claves públicas (PKI).
3. **Integridad de los datos:** Que la información llegue completa y sin corrupción en sus datos. De no ser así que sea posible detectarlo por cualquiera de las partes. Cierta integridad ya es proporcionada por la capa transporte mediante TCP.



***SSL***, es un servicio de propósito general que yace sobre TCP y da servicio de encriptamiento a la capa de aplicación. Podría ser ubicado en el nivel 6 de la arquitectura OSI (como los protocolos MIME, ASN.1), ya que cumple funciones de presentación al encriptar datos. A su vez también cumple con funciones del nivel 5 de la misma arquitectura ya que permite establecer sesiones, pausarlas y retomarla en otro momento además de aprovechar en una misma sesión el establecimiento de varias conexiones.

El uso de SSL no está restringido únicamente a HTTP (443). Otros protocolos también utilizan sus ventajas: FTPS (well known ports TCP 989, 990), TELNETS (992), IMAPS (993) y POP3S (995). La arquitectura de SSL especifica los siguientes protocolos:



***GET condicional***

HTTP proporciona un mecanismo que permite al cliente emplear la caché al tiempo que permite asegurar que todos los objetos pasados al navegador están actualizados. Un mensaje HTTP de petición es lo que se denomina un GET condicional si: 1) el mensaje de petición utiliza el método GET, y 2) el mensaje de petición incluye una línea de cabecera ***If-Modified-Since***:

1- un navegador demanda un objeto que no está en la caché:

GET /fruit/kiwi.gif HTTP/1.0

User-agent: Mozilla/4.0

2- el servidor web envía al cliente un mensaje de respuesta con el objeto:

HTTP/1.0 200 OK

Date: Web, 12 Aug 1998 15:39:29

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 09:23:24

Content-Type: image/gif

(cuerpo de entidad)

el cliente muestra el objeto al usuario, pero también guarda el objeto en su caché local. El cliente guarda la fecha de la última modificación junto con el objeto.

3- una semana más tarde el usuario pide el mismo objeto y el objeto está todavía en caché. como el objeto puede haber sido modificado en el servidor web durante la semana transcurrida, el navegador realiza una comprobación de actualización utilizando GET condicional. Específicamente el navegador envía:

GET /fruit/kiwi.gif HTTP/1.0

User-agent: Mozilla/4.0

If-modified-since: Mon, 22 Jun 1998 09:23:24

La línea de cabecera ***If-Modified-Since:*** es igual a la línea cabecera Last-Modified: enviada por el servidor una semana antes. El GET condicional india al servidor que envía el objeto solo si ha sido modificado desde la fecha especificada.

HTTP/1.0 304 Not Modified

Date: Web, 19 Aug 1998 15:39:29

Server: Apache/1.3.0 (Unix)

El servidor web sigue enviando un mensaje de respuesta, pero no incluye en él el objeto pedido. La inclusión del objeto demandado solo malgastaría el ancho de banda, e incrementa el tiempo de respuesta que percibe el usuario, particularmente si el objeto es grande.

***Contenido HTTP***

HTTP es utilizado también para transferir otros tipos de archivos. Se utiliza en aplicaciones de comercio electrónico para transferir archivos XML de una maquina a otra sin que ninguna de estas máquinas tenga un navegador o un usuario.

***Transferencia de archivos: FTP***

Para que el usuario pueda acceder a la cuenta remota debe, proporcionar una identificación de usuario y una palabra clave.

El usuario interactúa con FTP a través de un agente de usuario FTP.

Tanto HTTP como FTP son protocolos de transferencias de archivos que tienen características comunes, ambos funcionan sobre TCP.

La diferencia más notable consiste en que FTP utiliza dos conexiones TCP paralelas para transferir un archivo: ***una conexión de control*** y una ***conexión de datos***. La conexión de control se utiliza para enviar información de control entre las dos estaciones. La conexión de datos se utiliza para enviar un archivo, se dice que envía su información de control fuera de banda. HTTP envía las líneas de cabecera de petición y respuesta sobre la misma conexión TCP, envía su información de control en banda.

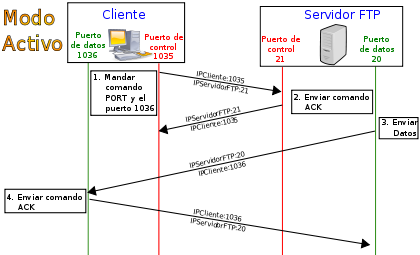
Cuando un usuario da comienzo a una sesión FTP con un host remoto, el lado cliente de FTP (usuario) inicia primeramente una conexión TCP de control con el lado del servidor sobre el puerto del servidor nro. 21. El lado del cliente de FTP envía la identificación de usuario a través de esta conexión de control.

FTP envía exactamente un archivo sobre la conexión de datos, y si durante la misma sesión, el usuario desea transferir otro archivo, FTP abre otra conexión de datos. La conexión de control permanece abierta mientras dure la sesión de usuario.

FTP debe mantener estado de usuario, HTTP no tiene estado.

### 

### Modo activo



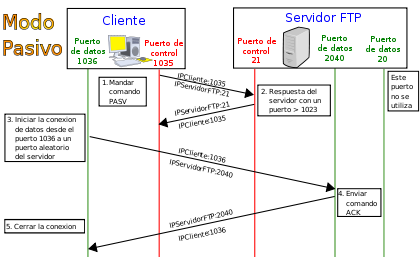
En modo Activo, el servidor siempre crea el canal de datos en su puerto 20, mientras que en el lado del cliente el canal de datos se asocia a un puerto aleatorio mayor que el 1024. Para ello, el cliente manda un comando PORT al servidor por el canal de control indicándole ese número de puerto, de manera que el servidor pueda abrirle una conexión de datos por donde se transferirán los archivos y los listados, en el puerto especificado.

Lo anterior tiene un grave problema de seguridad, y es que la máquina cliente debe estar dispuesta a aceptar cualquier conexión de entrada en un puerto superior al 1024, con los problemas que ello implica si tenemos el equipo conectado a una red insegura como Internet. De hecho, los [cortafuegos](https://es.wikipedia.org/wiki/Cortafuegos_(inform%C3%A1tica)) que se instalen en el equipo para evitar ataques seguramente rechazarán esas conexiones aleatorias. Para solucionar esto se desarrolló el modo pasivo.

### Modo pasivo

Cuando el cliente envía un comando PASV sobre el canal de control, el servidor FTP le indica por el canal de control, el puerto (mayor a 1023 del servidor. Ejemplo:2040) al que debe conectarse el cliente. El cliente inicia una conexión desde el puerto siguiente al puerto de control (Ejemplo: 1036) hacia el puerto del servidor especificado anteriormente (Ejemplo: 2040).[3](https://es.wikipedia.org/wiki/File_Transfer_Protocol#cite_note-3)

Antes de cada nueva transferencia tanto en el modo Activo como en el Pasivo, el cliente debe enviar otra vez un comando de control (PORT o PASV, según el modo en el que haya conectado), y el servidor recibirá esa conexión de datos en un nuevo puerto aleatorio (si está en modo pasivo) o por el puerto 20 (si está en modo activo). En el protocolo FTP existen 2 tipos de transferencia en ASCII y en binarios.



***Comandos y respuestas FTP***

los comandos y las respuestas son enviados sobre la conexión en formato ASCII de 7 bits.

***Correo electrónico en internet***

El correo electrónico es un medio asíncrono de comunicación, con el que las personas envías y reciben mensajes cuando quieren, sin tener que coordinarse con los horarios de otras personas.

**Tres componentes:**

Agentes de usuario, servidores de correo, y el protocolo simple de transferencia de correo (SMTP).

SMTP es el principal protocolo de la capa de aplicación para el correo electrónico. utiliza el servicio fiable TCP de transferencia de datos para enviar correo desde el servidor del correo del remitente hasta el del destinatario.

SMTP tiene dos lados: cliente, que ejecuta el servidor de correo del remitente y el lado servidor, que ejecuta el servidor del correo del destinatario. Cuando un servidor envía un correo a otros servidores, actúa como un cliente SMTP. Cuando un servidor recibe un correo de otros servidores actúa como un servidor SMTP.

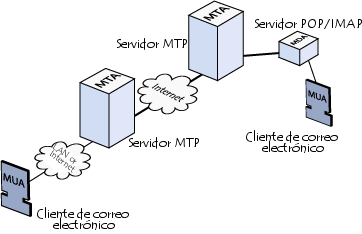
SMTP (RFC 282) Transfiere mensajes desde el servidor de correo de un remitente al de un destinatario.

Cuando se envía un correo electrónico, el mensaje se enruta de servidor a servidor (punto a punto) hasta llegar al servidor de correo electrónico del receptor. Más precisamente, el mensaje se envía al servidor del correo electrónico o MTA cliente, (Mail Transport Agent [Agente de Transporte de Correo]) que tiene la tarea de transportarlos hacia el MTA del destinatario. En Internet, los MTA se comunican entre sí usando el protocolo [SMTP](http://es.ccm.net/contents/internet/smtp.php3), y por lo tanto se los llama servidores SMTP (o servidores de correo saliente).

Luego el MTA del destinatario entrega el correo electrónico al servidor del correo entrante (llamado MDA, del inglés Mail Delivery Agent [Agente de Entrega de Correo]), el cual almacena el correo electrónico mientras espera que el usuario lo acepte. Existen dos protocolos principales utilizados para recuperar un correo electrónico de un MDA:

* [**POP**](http://es.ccm.net/contents/internet/smtp.php3) (Post Office Protocol [Protocolo de Oficina de Correo])
* [**IMAP**](http://es.ccm.net/contents/internet/smtp.php3)(Internet Message Access Protocol [Protocolo de Acceso a Mensajes de Internet])

Por esta razón, los servidores de correo entrante se llaman servidores POP o servidores IMAP, según el protocolo usado.

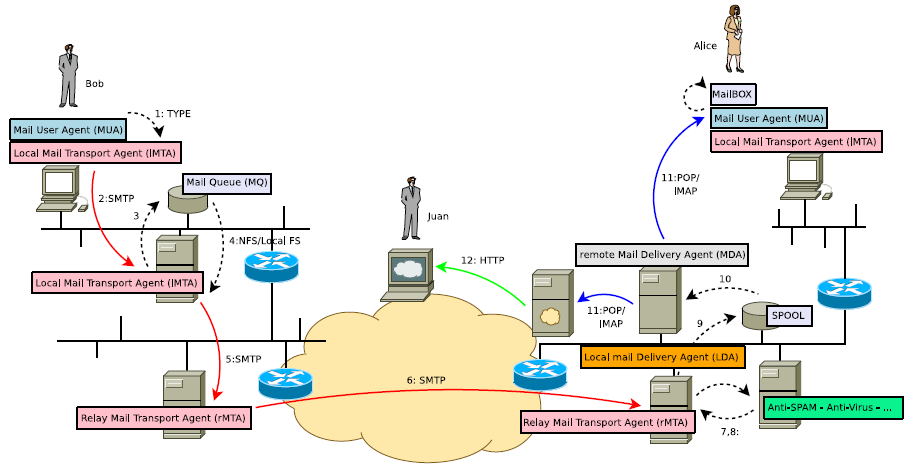


Usando una analogía del mundo real, los MTA actúan como la oficina de correo (el área de clasificación y de transmisión, que se encarga del transporte del mensaje), mientras que los MDA actúan como casillas de correo, que almacenan mensajes (tanto como les permita su volumen), hasta que los destinatarios controlan su casilla. Esto significa que no es necesario que los destinatarios estén conectados para poder enviarles un correo electrónico.

Para evitar que cualquiera lea los correos electrónicos de otros usuarios, el MDA está protegido por un nombre de usuario llamado registro y una contraseña.

La emisión, edición y recuperación del correo se logra a través de un programa de software llamado MUA (Mail User Agent [Agente Usuario de Correo]). Este posee un LMTA (Local Mail Transport Agent) para comunicarse con el servidor de correo saliente; utiliza los protocolos SMTP, POP, e IMAP para hablar con los MTA y los MDA.

Cuando el MUA es un programa instalado en el sistema del usuario, se llama cliente de correo electrónico (tales como Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredimail o Lotus Notes), pero cuando se usa una interfaz de web para interactuar con el servidor de correo entrante, se llama correo electrónico.

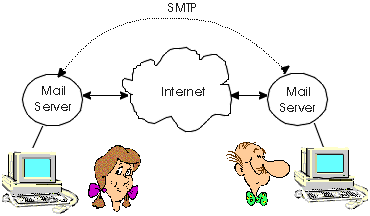


***Simple Mail Transfer Protocol (SMTP)***

Es un protocolo que puede actuar tanto como cliente o servidor, está definido en RFC 2821, y es el núcleo del correo electrónico de Internet. SMTP transfiere mensajes desde el servidor de correo de un remitente al de un destinatario. SMTP es mucho más antiguo que HTTP. A pesar de que SMTP tiene numerosas y excelentes cualidades, como lo demuestra su omnipresencia en Internet, tiene sin embargo una tecnología de legado que posee ciertas características arcaicas. Por ejemplo, la restricción a ASCII de siete bits del cuerpo (y no simplemente las cabeceras) de todos los mensajes de correo. Esta restricción tenía sentido a principios de la década de 1980, cuando la capacidad de transmisión era escasa y nadie enviaba grandes archivos adjuntos de imágenes, de audio o de vídeo. Pero en la actualidad, en la era multimedia, la restricción de ASCII de siete bits es un poco penosa; obliga a que los datos multimedia binarios sean codificados a ASCII antes de ser enviados sobre SMTP, y obliga a que el correspondiente mensaje ASCII sea descodificado de vuelta a binario después del transporte SMTP.

Para ilustrar la operativa básica de SMTP, revisemos un escenario común. Supongamos que Alicia desea enviar a Roberto un sencillo mensaje ASCII:

1. Alicia invoca a su agente de usuario para correo electrónico, proporciona la dirección de correo electrónico de Roberto (por ejemplo, roberto@escuela.edu), compone un mensaje, y ordena al agente de usuario que envíe el mensaje.
2. El LMTA del agente de usuario (MUA) de Alicia envía por SMTP el mensaje a su servidor de correo, donde es ubicado en una cola de mensajes.
3. El lado cliente de SMTP, ejecutándose en el servidor de correo de Alicia, ve el mensaje en la cola de mensajes. Abre una conexión TCP con un servidor SMTP, que está ejecutándose en el servidor de correo de Roberto.
4. Tras alguna sincronización SMTP inicial, el cliente SMTP envía el mensaje de Alicia sobre la conexión TCP.
5. En el servidor de correo de Roberto, el lado servidor de SMTP recibe el mensaje. Entonces, el servidor de correo de Roberto lo deposita en su buzón de correo.
6. Roberto invoca a su agente de usuario para leer convenientemente el mensaje.



Es importante observar que **SMTP** normalmente no utiliza servidores intermedios para enviar el correo, incluso aunque los dos servidores de correo estén localizados en lados opuestos del mundo. Si el servidor de Alicia está en Hong Kong y el de Roberto en Barcelona, la conexión TCP es directa entre los dos servidores. En particular, si el servidor de correo de Roberto está fuera de servicio, el mensaje permanece en el servidor de Alicia esperando a un nuevo intento (el mensaje no es albergado en ningún servidor de correo intermedio).

**SMTP utiliza conexiones persistentes:** si el servidor de correo emisor tiene que enviar diversos mensajes al mismo servidor receptor, puede enviarlos todos ellos a través de la misma conexión TCP. Para cada mensaje, el cliente comienza cada proceso con un nuevo MAIL FROM: crepes.fr. delimita el final de mensaje con un punto y realiza QUIT sólo después de que todos los mensajes hayan sido enviados.

**Post Office Protocol (POP)**, fue diseñado para ser fácil de implementar, pero tiene varias deficiencias y/o limitaciones en su funcionamiento. No maneja correctamente múltiples carpetas. Fuerza la descarga completa de los mensajes. En vez de descargar al ordenador del usuario sólo la cabecera “Asunto:” para que decida si el mensaje debe ser borrado o leído, descarga el mensaje completo con todos sus archivos adjuntos y una vez descargado es cuando el usuario puede procesarlo. Por último y lo más importante NO podremos procesar nuestro correo desde diferentes puestos (trabajo, casa, viajes,…), ya que descarga todos los mensajes desde la cola de entrada de las estafetas hasta el primer ordenador desde el que establecemos la conexión y por tanto, a partir de este momento, no será visible desde ningún otro.

**Internet Mail Acess Protocol (IMAP),** es un protocolo de acceso al correo, al igual que POP. Tiene muchas más funcionalidades que POP por lo que es más complejo. IMAP proporciona comandos que permiten a los usuarios crear carpetas y mover los mensajes de una carpeta a otro, IMAP también proporciona comandos para buscar en los buzones remotos mensajes que cumplen algún criterio en particular. Otra característica importante es que tiene comandos que permiten al agente de usuario obtener componentes (partes) de mensajes.

Fue diseñado para solucionar las carencias del protocolo POP, principalmente la movilidad y el procesamiento de correo desde diferentes puestos. 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.

Mediante este protocolo nuestro programa de correo electrónico conecta al servidor y descarga exclusivamente las cabeceras de los mensajes. Los mensajes quedan almacenados en el “buzón del usuario”. Al descargar solamente las cabeceras (y no todo el cuerpo del mensaje) podemos eliminar los mensajes no deseados sin necesidad de descargar el mensaje en su totalidad; los mensajes solo son transferidos cuando se seleccionan individualmente para leerlos.

Este protocolo, sin embargo, obliga a mantener abierta la conexión hasta finalizar la consulta y por tanto no es aconsejable cuando disponemos de una conexión con un alto coste por minuto y no permanente.

Lo más interesante del protocolo IMAP es que al quedar los buzones y su contenido situados físicamente en los repositorios centrales y no en el PC del usuario, nos permite acceder a los mensajes desde diferentes ordenadores, aunque previamente los hayamos leído desde otra máquina o programa. Es especialmente interesante en desplazamientos fuera del entorno de trabajo pues los buzones y mensajes podrán leerse mediante el servicio WEBMAIL durante el desplazamiento y posteriormente desde el cliente de correo habitual.

***Diferencias entre el servicio ofrecido por POP vs el ofrecido por IMAP?***

POP e IMAP son los dos protocolos de recepción de correo electrónico más utilizados, que soportan la mayoría de los servidores de correo electrónico y programas clientes como Outlook Express o Mozilla Thunderbird. También podemos referirnos a ellos con sus números de versión POP3 e IMAP4.

Existen varias diferencias, pero digamos que IMAP es más novedoso y permite más funcionalidades.

La diferencia fundamental es que, cuando nos conectamos por POP, se descargan todos los mensajes al ordenador cliente y con IMAP no se descargan todos mensajes sino sólo los que el cliente solicite. Además, lo normal cuando se accede a un correo con POP es que los mensajes se borren del servidor según se descargan, mientras que por IMAP al visualizar un mensaje no se borra del servidor, a no ser que se elimine explícitamente. Otras diferencias es que con POP sólo se puede conectar un usuario para descargar el correo electrónico y con IMAP se pueden conectar más de un usuario a la misma cuenta. En cuanto a lo más recomendable, si POP o IMAP, eso depende del uso que des a tu correo y el modo de conexión.

Con POP puedes descargar todos los mensajes en tu ordenador en un pequeño espacio de tiempo y luego puedes verlos aunque no esté conectado a Internet. Con POP puedes escribir todas las respuestas sin estar conectado y luego volver a conectarte a Internet un momento para enviar las respuestas.

Con IMAP tienes que estar conectado a Internet todo el tiempo que leas tu correo y contestes los mensajes. Si pierdes la conexión a Internet no podrás acceder a tu correo recibido, porque está almacenado en el servidor de correo y no en tu ordenador.

La parte buena de IMAP es que varias personas pueden estar conectadas a una misma cuenta a la vez. Además, si cambias de ordenador en cualquier momento, podrás acceder a tus mensajes igualmente, porque por IMAP todos los mensajes están disponibles desde cualquier ordenador conectado a Internet.

Por lo tanto, con POP si descargas los mensajes, se guardarán en tu ordenador y tendrás que tener tu ordenador para leer el correo antiguo. Si los miras por IMAP seguirán almacenándose en el servidor hasta que los borres. Por eso IMAP puede ser una buena idea si cambias de ordenador habitualmente.

***Relacione DNS con SMTP.***

DNS básicamente sirve de soporte para el resto de los protocolos de aplicación como SMTP. Al escribir un mail, el user agent busca la IP del servidor de correos del dominio que ofrece la cuenta (servidor SMTP de yahoo.com) para realizar la conexión SMTP y enviar el correo. El servidor de yahoo, utiliza luego DNS para averiguar la IP del servidor de correo de hotmail.com, establece la conexión y envía el email. El destinatario, eventualmente revisa su correo con su user agent, que puede o no utilizar DNS para averiguar la IP del servidor POP3 o IMAP.

El envío de un mensaje de correo es un proceso largo y complejo. Éste es un esquema de un caso típico:

***Multipurpose Internet Mail Extensions (MIME)***

(en español "extensiones multipropósito de correo de internet") son una serie de convenciones o especificaciones dirigidas al intercambio a través de [Internet](https://es.wikipedia.org/wiki/Internet) de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos. En sentido general las extensiones de MIME van encaminadas a soportar:

* Texto en conjuntos de caracteres distintos de [US-ASCII](https://es.wikipedia.org/wiki/C%C3%B3digo_ASCII);
* adjuntos que no son de tipo texto;
* cuerpos de mensajes con múltiples partes (multi-part);
* información de encabezados con conjuntos de caracteres distintos de ASCII.

Prácticamente todos los mensajes de correo electrónico escritos por personas en [Internet](https://es.wikipedia.org/wiki/Internet) y una proporción considerable de estos mensajes generados automáticamente son transmitidos en formato MIME a través de [SMTP](https://es.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol). Los mensajes de correo electrónico en Internet están tan cercanamente asociados con el SMTP y MIME que usualmente se les llama mensaje SMTP/MIME.

MIME está especificado en seis [Request for Comments](http://comments) o RFC (en español "solicitud de comentarios): [RFC 2045](https://tools.ietf.org/html/rfc2045), [RFC 2046](https://tools.ietf.org/html/rfc2046), [RFC 2047](https://tools.ietf.org/html/rfc2047), [RFC 4288](https://tools.ietf.org/html/rfc4288), [RFC 4289](https://tools.ietf.org/html/rfc4289) y [RFC 2077](https://tools.ietf.org/html/rfc2077).

En la actualidad ningún programa de [correo electrónico](https://es.wikipedia.org/wiki/Correo_electr%C3%B3nico) o [navegador](https://es.wikipedia.org/wiki/Navegador_web) de Internet puede considerarse completo si no acepta MIME en sus diferentes facetas (texto y formatos de archivo).

Cuando una persona envía un correo electrónico a otra, se incluye una cabecera que contiene información periférica y que precede al propio cuerpo del mensaje. Esta información periférica está contenida en una serie de líneas de cabecera definidas en la RFC 822. RFC 822 especifica el formato exacto de las líneas cabeceras del correo, así como sus interpretaciones semánticas.

From: [alicia@crepes.fr](mailto:alicia@crepes.fr)

To: [roberto@hamburger.edu](mailto:roberto@hamburger.edu)

Subject: buscando el significado de la vida

***SMTP Comparado con HTTP***

Ambos protocolos son usados para transferir archivos de un host a otro. HTTP transfiere archivos desde un servidor a un cliente web; SMTP transfiere archivos de un servidor de correo a otro. cuando se transfieren los archivos, tanto HTTP persistentes como SMTP, utilizan conexiones persistentes. Por esto los dos protocolos tienen características comunes.

Las diferencias: 1) HTTP es un protocolo de demanda, la conexión TCP es iniciada por la maquina que quiere recibir el archivo. SMTP es un protocolo de oferta, el servidor de correo emisor envía el archivo al servidor de correo receptor. La conexión TCP es iniciada por la maquina que quiere enviar el archivo.

2) SMTP requiere que cada mensaje, incluyendo sus cuerpos, este en formato ASCII de siete bits. Los datos en HTTP no tienen esa restricción.

3) La forma de gestionado de un documento que consta de texto e imágenes. HTTP encapsula cada objeto en su propio mensaje HTTP de respuesta. SMTP reúne todos los objetos en un mensaje.

***UNIDAD 3 – Capa de Transporte***

La capa de transporte proporciona una comunicación lógica entre procesos de aplicación que se ejecutan sobre máquinas diferentes.

Los procesos de aplicación utilizan la comunicación lógica proporcionada por la capa de transporte para enviarse mensajes entre ellos, despreocupándose de los detalles de la infraestructura física utilizada para llevar esos mensajes. Los protocolos de mensajes disponibles para las aplicaciones TCP y UDP.

***Estructura del segmento TCP y UDP***

***Estructura del segmento UDP:***

|  |  |
| --- | --- |
|  | Puertos: MUX/DEMUX.  Longitud: UDP HDR + Payload.  Checksum  • Cálculo Ca1, Opcional. 0 = Sin Checksum.  • Calculado HDR + PseudoHDR + Payload.  • Aplicaciones de LAN por eficiencia lo podrían deshabilitar.  • Si tiene error se descarta silenciosamente. |

La cabecera UDP solo tiene cuatro campos, cada uno de dos bytes, los números de puertos permiten que en el host destino pasen los datos de aplicación al proceso adecuado que se ejecuta en el sistema final.

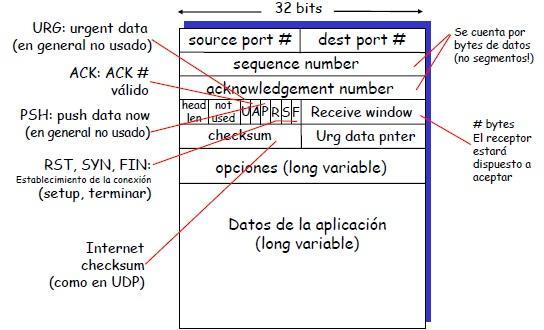
La suma de comprobación se utiliza en el host receptor para comprobar si ha introducido algún error en el segmento.

El campo de longitud especifica la longitud de segmento UDP en bytes incluyendo la cabecera.

***Estructura del segmento TCP:***

Los campos de puerto origen y puerto destino identifican los puntos terminales locales de la conexión. El campo Longitud de cabecera indica la cantidad de palabras de 32 bits contenidas en el header TCP. Esta información es necesaria porque el campo de opciones es de longitud variable, por lo que el header también. Reservado es un campo de 6 bits que no se usa. Luego vienen seis flags de 1 bit:

* **URG** se establece en 1 si está en uso el apuntador urgente. El apuntador urgente sirve para indicar un desplazamiento en bytes a partir del número actual de secuencia en el que se encuentran datos urgentes. Este recurso sustituye los mensajes de interrupción.
* **ACK** se establece en 1 para indicar que el número de acuse de recibo (acknowledgement) es válido. Si el ACK es 0 el segmento no contiene un acuse de recibo, por lo que se ignora el campo de acuse de recibo.



* **PSH** indica datos empujados (con push). Por este medio se solicita atentamente al receptor entregar los datos a la aplicación a su llegada y no ponerlos en buffer hasta la recepción de un buffer completo.
* **RST** para restablecer una conexión que se ha confundido debido a una caída de host u otra razón; también sirve para rechazar un segmento no válido o un intento de abrir una conexión.
* **SYN** se usa para establecer conexiones. La solicitud de conexión tiene SYN = 1 y ACK = 0 para indicar que el campo de acuse de recibo incorporado no está en uso. La respuesta de conexión sí lleva un reconocimiento, por lo que tiene SYN = 1 y ACK= 1. En esencia, el bit SYN se usa para denotar conection request y conection acepted, usándose el bit ACK para distinguir entre ambas posibilidades.
* **FIN** se usa para liberar una conexión; especifica que el transmisor no tiene más datos que transmitir. Sin embargo, tras cerrar una conexión, un proceso puede continuar recibiendo datos indefinidamente. Ambos segmentos, SYN y FIN tienen números de secuencia y por tanto tienen la garantía de procesarse en el orden correcto.

El tamaño de la ventana indica cuantos bytes se pueden mandar después del byte de acuse de recibo. El valor 0 es legal; Uno puede dar permiso para transmitir de nuevo con un segmento con el mismo número de acuse y una ventana más de longitud 0. El checksum es una suma de comprobación similar a la de UDP. Los campos de número de secuencia y número de asentamiento de recibo desempeñan sus funciones normales. Hay que notar que el segundo campo especifica el siguiente byte esperado, no el último byte correctamente recibido.

***Objetivo del uso de puertos en el modelo TCP/IP***

Tiene como objetivo la conexión de redes múltiples y la capacidad de mantener conexiones aun cuando una parte de la subred está perdida.

Familia de protocolos sobre los cuales funciona Internet, que se ha convertido en el estándar actual de comunicación entre computadoras. **IP, que se ocupa de transferir los paquetes de datos hasta su destino adecuado y el protocolo TCP, que se ocupa de garantizar que la transferencia se lleve a cabo de forma correcta y confiable.**

El uso de puertos en el modelo TCP/IP permite la multiplexación (combinación de dos o más [canales de información](http://comunicaciones) en un solo [medio de transmisión](http://transmisión)) de los paquetes que la capa de transporte recibe de los distintos procesos y pasa a la capa de Red en el emisor, y la demultiplexación de los segmentos recibidos desde la capa de Red en paquetes que se envían a los procesos adecuados en el receptor.

Si un terminal tuviese dos procesos haciendo solicitudes a un mismo servidor Web, ambos se comunicarían al port 80 (utilizado por el servidor Web en el host destino).  
 El servidor Web se “forkea” (ramificación), es decir crea un nuevo proceso (hilo) para atender cada nueva solicitud (en este ejemplo tendría dos procesos) y así tener el proceso central escuchando el port 80. El port destino permite a la capa de Transporte del receptor entregar los paquetes (demultiplexación) al servidor Web y no a otro proceso que esté corriendo en ese host (servidor). Del mismo modo, cada proceso del emisor tendrá asociado un port de origen diferente (comúnmente aleatorio), permitiendo al servidor Web encaminar sus respuestas en dirección contraria a un proceso específico del host (cliente) que realizó las solicitudes.

En el caso que dos clientes (distintos, es decir, diferentes host) escojan un mismo port origen, el servidor podrá diferenciar a cada proceso utilizando la IP de origen, en el datagrama de capa de Red. Así, la tripla (port origen, port destino, IP origen) permite al host receptor encaminar los datos al socket correspondiente.

**Compare TCP y UDP en cuanto a: Confiabilidad, Multiplexación, Orientado a la conexión, Controles de congestión y Utilización de puertos.**

La capa de Transporte ofrece dos protocolos, cada uno con distintos servicios: TCP y UDP. Por estar construidos sobre el único protocolo que ofrece la capa de Red, IP, algunos servicios no pueden ser ofrecidos por ninguno de ellos: garantías de retardo ni de ancho de banda -básicamente porque IP utiliza conmutación de paquetes y no de circuitos-. Sin embargo, otras características no soportadas por IP, como la transmisión confiable, sí pueden ser ofrecidas por la capa de Transporte.

|  |  |  |  |
| --- | --- | --- | --- |
| Característica | Definición | TCP | UDP |
| Confiabilidad | Asegurar la entrega de todos los paquetes enviados, permitiendo mantener en el receptor el orden en que le fueron enviados (utilizando confirmación de recepción, timeout, reenvío de paquetes y números de secuencia) y detectar errores en la recepción (mediante *checksum*). | SI (utiliza establecimiento de conexión de 3 pasos) | NO (sin conexión, sólo ofrece el *best effort* de IP) (\*) |
| Multiplexación | Extender la comunicación host-host que ofrece IP permitiendo diferenciar los procesos en el receptor y en el emisor, de modo que varios procesos en un host se comuniquen con varios procesos en otros hosts, sin confundir los paquetes (utilizando direcciones IP y números de puerto que identifican a los procesos). | SI Socket: (*IP Orig, Port Orig, IP Dest, Port Dest*) | SI Socket: (*IP Dest, Port Dest*) (\*\*) |
| Control de Congestión | Controlar la velocidad de transmisión del emisor para no sobrecargar la red. | SI | NO |
| Control de Flujo | Controlar la velocidad de transmisión del emisor para no sobrecargar al receptor (según el tamaño de su buffer, ventana de recepción). | SI | NO |
| Utilización de puertos | Los ports destino y origen son precisos tanto en TCP como en UDP por fines de multiplexación/demultiplexación, aunque UDP sólo precisa el port destino para definir un socket (al ser sin conexión se envían segmentos individuales, mientras que TCP divide un mensaje en múltiples segmentos y precisa reconocer el proceso emisor para recomponer el mensaje original). | Port Origen y Port Destino | Port Origen y Port Destino |
| Orientado a Conexión | Si se establece una conexión previa entre los host (sincronización), con el fin de establecer un canal seguro de comunicación. | SI | NO |

(\*) Es posible un Transporte Confiable Sobre UDP -TCsU-, pero dependiente de la aplicación -será esta quien determine cómo recuperarse ante fallos-. Del mismo modo que TCP ofrece transmisión confiable sobre IP, una aplicación puede lograr transmisión confiable sobre UDP. Si bien podría utilizarse TCP en estos casos y simplificar la aplicación, el uso de TCsU ofrece a la aplicación una ventaja: UDP no posee control de congestión, de modo que puede transmitir a cualquier tasa de transferencia aunque perjudique el desempeño total de la red -con TCP, se comienza enviando datos a una tasa baja que aumenta gradualmente; por cada segmento no confirmado se baja la tasa por detección de congestión y en conjunto con IP se logra un control de congestión preventivo.  
  
 (\*\*) UDP: Segmentos con distinto IP y port de origen pueden ser dirigidos a un mismo socket, dado que sólo lo determina el IP y port destino. El port origen provee una dirección de retorno, por si el receptor desea responder de algún modo al emisor, pero esto dependerá del protocolo de aplicación, no es propio de UDP, pero TCP en cambio, envía siempre un paquete de acuse de recibo o ACK.

***La PDU de la capa de transporte es el segmento. Sin embargo, en algunos contextos suele utilizarse el término Datagrama, indique cuándo.***

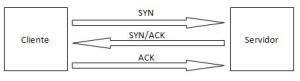
El termino datagrama para la PDU de capa de transporte se utiliza cuando usamos UDP.

Para referirse a la 4-PDU se utiliza el término segmento. También se menciona el término segmento para referirnos a la PDU para TCP. El termino datagrama hace referencia a la PDU para UDP. También se usa el mismo término para la PDU de la capa de Red.

**Saludo de tres vías de TCP**

* **Paso 1**: Host cliente envía segmento TCP SYN al servidor
  + Especifica # de secuencia inicial
  + No data
* **Paso 2**: Host servidor recibe el SYN, responde con segmento SYN & ACK
  + Servidor ubica buffers
  + Especifica el # de secuencia inicial
* **Paso 3**: el cliente recibe SYN & ACK, responde con segmento ACK, el cual podría contener datos.

El cliente realiza una conexión enviando un paquete SYN al servidor, en el servidor se comprueba si el puerto está abierto (si existe un proceso escuchando por ese puerto), si el puerto no está abierto se le envía al cliente un paquete de respuesta RST, esto significa un rechazo de intento de conexión. Si el puerto está abierto, el servidor responde con un paquete SYN/ACK. Entonces el cliente respondería al servidor con un ACK, completando así la conexión.



**ACK** o “acknowledge” (1 bit): Si está activo entonces el campo con el número de acuse de recibo es válido (si no, es ignorado). Es un mensaje que se envía para confirmar que un mensaje o un conjunto de mensajes han llegado.

SYN o “synchronize” (1 bit): Activa/desactiva la sincronización de los números de secuencia. Se usa para sincronizar los números de secuencia en tres tipos de segmentos: petición de conexión, confirmación de conexión (con ACK activo) y la recepción de la confirmación (con ACK activo).

RST o “reset” (1 bit): Si llega a 1, termina la conexión sin esperar respuesta. Es un bit que se encuentra en el campo del código en el protocolo TCP, y se utiliza para reiniciar la conexión. Un ejemplo práctico de utilización es el que realiza un servidor cuando le llega un paquete a un puerto no válido: este responde con el RST activado.

***Multicast***

Multidifusión (multicast) es el envío de la información en una [red](http://es.wikipedia.org/wiki/Red_de_computadoras) a múltiples destinos simultáneamente, usando la estrategia más eficiente para el envío de los mensajes sobre cada enlace de la red sólo una vez y creando copias cuando los enlaces en los destinos se dividen. En oposición a multicast, los envíos de un punto a otro en una red se denominan [unidifusión](http://es.wikipedia.org/wiki/Unicast) ([unicast](http://es.wikipedia.org/wiki/Unicast)), y los envíos a todos los nodos en una red se denominan [difusión amplia](http://es.wikipedia.org/wiki/Broadcast_(inform%C3%A1tica)) ([broadcast](http://es.wikipedia.org/wiki/Broadcast_(inform%C3%A1tica))).

Funciona sobre el protocolo UDP, no se podría adaptar a TCP porque es unicast.

El multicast está orientado hacia aplicaciones del tipo "uno para muchos" y "muchos para muchos". En estos casos, presenta claras ventajas cuando se lo compara con los mecanismos de transmisión unicast y broadcast. En unicast, es necesario que la fuente replique varios flujos de datos idénticos con el objeto de transmitirlos a cada uno de los receptores, generando desperdicio de banda. Por otro lado, el sistema broadcast envía los datos a toda la red de forma indiscriminada. Esto también da como resultado el desperdicio de recursos, pues implica el transporte de datos para todas las estaciones de la red, aunque el número de receptores deseosos de que ese contenido sea reducido. Con multicast, la fuente de tránsito envía una única copia de los paquetes hacia una dirección de grupo multicast. La infraestructura de red replica estos paquetes de forma inteligente, encaminando los datos de acuerdo con la topología de receptores interesados en esa información.

Entre las diversas aplicaciones que pueden obtener ganancias con el uso de multicast están: videoconferencia; aprendizaje a distancia; distribución de software, noticias e informaciones de mercado; conciertos al vivo; actualización de bases de datos; juegos distribuidos; procesamiento competidor; simulacros distribuidos, etc.

TCP es orientado a conexión, el establecimiento de la misma de tres vías es uno a uno (punto a punto). Con UDP si es posible multicast.  
 No es posible establecer comunicaciones multicast sobre TCP, las mismas se realizan sobre UDP debido a que no son comunicaciones punto a punto. Con TCP antes de comenzar el envío de datos, debería establecerse una comunicación mediante el handshake de tres vías. Ese establecimiento de conexión es con una cardinalidad uno a uno, lo que no resultaría práctico para el establecimiento de conexión uno a muchos (lo que es, en forma sencilla, la idea de multicast). Por esto, se utiliza UDP para multicasting, debido a que no es necesario un establecimiento de conexión previo al envío de datos.

***TFTP “Trivial File Transfer Protocol”***

Es un protocolo de transferencia de archivos muy simple, que funciona sobre UDP puerto #69 en vez de TCP #20 y #21 como FTP. Al no ser orientado a conexión, no tiene autentificación de usuario, encriptación, y no puede listar contenidos de directorios.  
 Ya que TFTP utiliza UDP, no hay una definición formal de sesión, cliente y servidor, aunque se considera servidor a aquel que abre el puerto 69 en modo UDP, y cliente a quien se conecta.  
Sin embargo, cada archivo transferido vía TFTP constituye un intercambio independiente de paquetes, y existe una relación cliente-servidor informal entre la máquina que inicia la comunicación y la que responde.

En cuanto a los mensajes intercambiados: Con TFTP no hay establecimiento de conexión ni confirmaciones. Los mensajes fueron: pedido de escritura del cliente, acuse de recibo del servidor, envío del archivo desde el cliente en bloque #1, acuse de recibo del bloque #1 desde el servidor, pedido de lectura, bloque de datos #1 desde el servidor, acuse de recibo del bloque #1 desde el cliente.

***ARQ (Automatic Repeat Request)***

El **ARQ** es un protocolo utilizado para el control de errores en la transmisión de datos, garantizando la integridad de los mismos. Este suele utilizarse en sistemas que no actúan en tiempo real ya que el tiempo que se pierde en el reenvió puede ser considerable y ser más útil emitir mal en el momento que corresponde un tiempo después. Esto se puede ver muy claro con una aplicación de videoconferencia donde no resulta de utilidad emitir el pixel correcto de la imagen d segundos después de haber emitido la imagen.

Esta técnica de control de errores se basa en el reenvió de los paquetes de información que se detecten como errores.

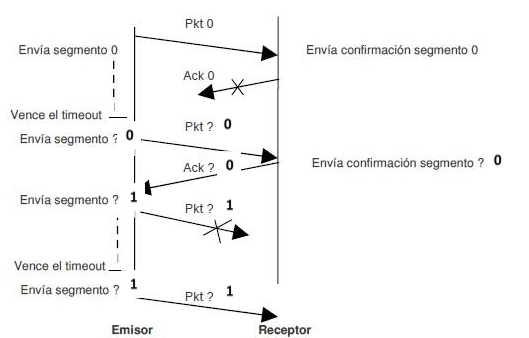
Para controlar la correcta recepción de un paquete se utilizan **ACK** y **NACK** si no es correcto. Durante el protocolo que controla la recepción de paquetes pueden surgir múltiples problemas complicándose así el contenido del ACK y surgiendo nuevos conceptos como el de timeout.

Si el emisor no recibe información sobre la recepción del paquete durante un tiempo fijado este se reenvía automáticamente.

**ARQ** requiere de tres capacidades adicionales para manejar la presencia de bits erróneos:

* *Detección de errores*: el receptor debe detectar que se ha producido un error de bit
* *Realimentación del receptor*: el receptor le avisa al emisor que el paquete llego.
* *Retransmisión*: todo paquete que ha sido recibido con error serán retransmitido peor el emisor.

Algunos protocolos ARQ son: *Stop-And-Wait, Go-Back-N y Selective Repeat*.



**Stop-and-Wait:** el emisor envía un paquete y queda en el estado de espera de un ACK. Cuando el emisor esta en éste estado no puede conseguir más datos desde la capa superior (capa de aplicación) hasta que reciba un ACK y deje este estado, o se venza el tiempo de espera de respuesta (timeout).

Sólo hay en cada momento un segmento enviado esperando confirmación. El rango de los números de secuencia es de sólo un bit [0;1], por lo que es un protocolo sin *pipelining*.

Este es un protocolo con muchas deficiencias de performance y de prestaciones. La baja tasa de de transmisión de datos, hace que se desaproveche el potencial del hardware sobre el que se transportan los mismos, por lo que no es muy utilizado.

Como solución a esto, se pueden considerar dos técnicas básicas para la recuperación de errores en el entubado: Retroceder N (Go Back-N) y repetición selectiva (Selective Repeat).

***Go Back – N.:*** Es un protocolo de tipo Retroceder N (Go Back - N), el emisor puede transmitir varios paquetes sin esperar ningún reconocimiento, pero tiene restringido el nº máximo N de paquetes no reconocidos en el entubado.

El rango de números de secuencias permitido para los paquetes transmitidos pero aun no reconocidos puede verse como una ventana de tamaño N sobre el rango de nº de secuencias. Esta ventana se desliza avanzando sobre el espacio de nº de secuencias.

Este protocolo se suele llamar protocolo de ventana deslizante.

* **Acción del emisor:**

1. **Invocación desde arriba:** cuando quiere enviar un paquete, 1º comprueba si la ventana está llena (si hay paquetes pendientes de reconocimiento). Si *la ventana ni está llena*, se crea un paquete y se envía, pero *si está llena*, el emisor devuelve los datos hacia la capa superior indicando que la ventana está llena. La capa superior tendrá que intentarlo de nuevo.
2. **Recepción de un ACK:** un reconocimiento de paquete con un nº de secuencia **n**, será tomado como un reconocimiento acumulativo que indica que han sido recibidos correctamente en el receptor, todos los paquetes con numero de secuencias hasta el **n** incluido.
3. **Un evento de fin de tiempo límite de espera:** se utilizara un temporizador para recuperarse de la pérdida de paquetes que han sido previamente enviados pero que aun no han sido conocido.

* **Acción del receptor:**

Si se recibe correctamente y en orden el paquete con el nº de secuencia N, envía un ACK para el paquete N y entrega la porción de datos del paquete a la capa superior. En otros casos descarta el paquete y reenvía el ACK correspondiente al paquete mas reciente recibido en orden. Descarta paquetes que no lleguen en orden, porque no posee capacidad de almacenamiento y además por la configuración del protocolo GBN, ese paquete será reenviado nuevamente.

***Selective Repeat:*** Resuelve el problema de GBN de que el emisor llene la tubería con paquetes, existen, sin embargo, escenarios en los que el mismo GBN puede padecer problemas de prestaciones, en concreto, cuando el tamaño de ventana y el retardo debido al ancho de banda resultan grandes. Un simple error en un paquete puede ocasionar que GBN retransmita un gran número de paquetes (debido a un gran tamaño de la ventana), muchos de los cuales resultan innecesarios, con lo que el dictado de paquetes se vería “retardado” por todas estos reenvíos. Según se incremente la probabilidad de error ir en el canal, la tubería se puede llenar de retransmisiones innecesarias.

***Selective Repeat*** (SR) es otro protocolo ARQ con *pipelining*. Éste evita la retransmisión innecesaria de paquetes haciendo que el emisor retransmita únicamente aquellos paquetes sospechosos de haber sido recibidos con errores (los que fueron perdidos o corrompidos) por el receptor. Esta retransmisión individual, requerirá que el receptor reconozca los paquetes recibidos de forma *individualizada*. Tanto el emisor como el receptor manejan una ventana iguales de tamaño N para limitar el número de paquetes pendientes de reconocimiento en la tubería, enviando el receptor ACK's individuales para cada segmento. Esto evita retransmisiones innecesarias, dado que el receptor no descarta tantos segmentos y puede pedir la retransmisión de uno específico.

La restricción que existe sobre el tamaño de ventanas en este protocolo son:

* si el time out es muy grande hay problemas
* el receptor deber tener un buffer muy grande
* el receptor no puede indicar al emisor que está en problemas.

Este tipo de ARQ exige una memoria en el transmisor que sea capaz de almacenar tantos datos como los que puedan enviarse en un timeout, ya que será el tiempo máximo de espera y esos datos deben reenviarse tras detectar un error.

Otra de las exigencias de este tipo de ARQ es la numeración de los ACK’s para poder distinguir a que paquete de información están asistiendo.

La recepción desordenada de la información, lo que obliga a mantener otra ventana en recepción para poder pasar los datos de manera ordenada a la capa superior en caso de recibir un paquete con errores.

***Netstat (network statistics)*** es una herramienta de línea de comandos que muestra un listado de las conexiones activas de un ordenador, tanto entrantes como salientes. La información que resulta del uso del comando incluye el protocolo en uso, las direcciones IP tanto locales como remotas, los puertos locales y remotos utilizados y el estado de la conexión.

t: TCP  
u: UDP  
a: all  
n: numeric  
p: program

Puertos:

* Un servidor Web: 80/TCP
* Un servidor SSH: 22/TCP
* Un servidor DNS: 53/TCP
* Un servidor Web Seguro: 443/TCP
* Un servidor POP3: 110/TCP
* Un servidor IMAP: 143/TCP
* Un servidor SMTP: 25/TCP

***Programación sobre Sockets***

El núcleo de una aplicación de red consta de un par de programas: ***el programa cliente y el programa servidor***. Cuando se ejecutan estos dos programas, se crea un proceso cliente y un proceso servidor, y estos dos procesos se comunican entre sí por medio de mensajes (leyendo y escribiendo) a sockets. Cuando se crea una aplicación de red, la principal tarea del desarrollador es escribir el código de los dos programas cliente y servidor.

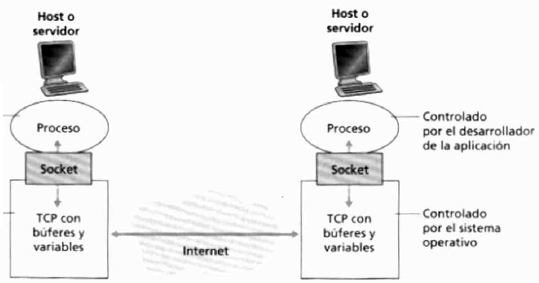
Existen dos tipos de aplicación cliente-servidor. Un tipo de aplicación es la aplicación cliente servidor que es una implementación de un protocolo estándar definido en una RFC. Para esa implementación, los programas cliente y servidor deben ser conformes con las reglas descritas en la RFC.

Si un desarrollador escribe el código del programa cliente, un desarrollador independiente escribe el código del programa servidor, y ambos siguen cuidadosamente las reglas de la RFC, entonces los dos programas deberán ser capaces de interpretar. De hecho, la mayoría de las aplicaciones de red de hoy en día involucran la comunicación entre programas cliente servidor que han sido creados por desarrolladores independientes

Cuando un programa cliente o servidor implementa un protocolo definido en una RFC, debe utilizar el número de puerto asociado con el protocolo.

El otro tipo de aplicaciones cliente-servidor es el de las aplicaciones propietarias. En este caso, los programas cliente y servidor no tienen necesariamente que seguir ninguna RFC existente. Un único desarrollador (o equipo de desarrollo) crea ambos programas cliente-servidor, teniendo completo control sobre lo que va en el código. Dado que el código no implementa ningún protocolo de dominio público, otros desarrolladores independientes no serán capaces de construir código que interopere con la aplicación. Cuando se desarrolla una aplicación propietaria, el desarrollador debe tener cuidado de no utilizar uno de los bien conocidos números de puerto definidos en las RFC.

Durante la fase de desarrollo de aplicaciones propietarias, una de las primeras decisiones que debe tomar el desarrollador es si la aplicación va sobre TCP o sobre UDP. Recuérdese que TCP es orientado a la conexión, y que proporciona un canal fiable de flujo de bytes, a través del cual fluyen los datos entre dos sistemas finales. UDP es sin conexión, y envía paquetes independientes de datos entre dos sistemas finales, sin ninguna garantía de entrega.



El desarrollador de la aplicación tiene control completo sobre el lado de la capa de aplicación del socket, pero tiene un control pequeño sobre el lado de la capa de transporte. (Como mucho, el desarrollador de la aplicación puede fijar algunos pocos parámetros TCP. como el tamaño máximo de búfer y de segmento).

El cliente se encarga de iniciar el contacto con el servidor. Para que el servidor pueda reaccionar al contacto inicial del cliente, el servidor tiene que estar preparado. Esto implica dos cosas. La primera es que el programa servidor no puede estar inactivo; debe estar ejecutándose como un proceso antes de que el cuente intente iniciar el contacto. La segunda es que el programa servidor debe tener algún tipo de puerta (esto es el socket) que reciba el contacto inicial del cliente que se ejecuta sobre una máquina arbitraria.

***Multiplexación y demultiplexación en la capa de transporte***

En el host destino, la capa de transporte recibe segmentos (esto es. las PDU de la capa de transporte) desde la capa de red que se encuentra justo debajo. La capa de transporte tiene la responsabilidad de entregar los datos almacenados en los segmentos a los procesos de aplicación apropiados que corren en los host. Veamos un ejemplo de esto. Suponga que está usted sentado delante de su computador, bajándose páginas web mientras tiene abierta una sesión de FTP y dos de Telnet. Por lo tanto, usted tiene cuatro aplicaciones de red ejecutándose (dos procesos Telnet, un proceso FTP y otro proceso HTTP). Cuando la capa de transporte de su computador recibe datos de la capa de red subyacente, necesita conducir los datos recibidos hasta uno de esos cuatro procesos.

Un proceso (como parte de una aplicación de red) tiene un socket, que es por donde que pasan los datos de la red al proceso y del proceso a la red. De este modo, la capa de transporte del host receptor no entrega los datos al proceso directamente, sino que se los entrega a un socket que actúa como intermediario. Como en un instante dado puede haber más de un socket abierto en el host receptor, cada socket tiene un identificador único.

Consideremos ahora un host que recibe directamente un segmento de la capa de transporte en el socket apropiado. Cada segmento de la capa de transporte tiene un conjunto de campos en el segmento para este propósito (puerto origen y puerto destino). En el extremo receptor, la capa de transporte examina estos campos (puerto destino) para identificar el socket receptor, y entonces dirigir el segmento a ese socket. Esta tarea de entregar los datos de un segmento de la capa de transporte al socket adecuado se denomina demultiplexión. Tener en cuenta esta demultiplexción para el caso de que haya varios procesos que en un momento dado tenga un socket conectado a la red y con segmentos de entrada, por lo que esta actividad se debe realizar en conjunto con el fin de dirigir los segmentos de información al socket del proceso correspondiente.

Por otro lado, el trabajo de reunir los paquetes de datos de los diferentes sockets (pertenecientes a uno o más procesos) en el host origen, encapsularlos con cabeceras de información (que será después utilizada para demultiplexarlos) para crear segmentos, y pasar estos segmentos a la capa de red se denomina multiplexado.

Entonces, sabemos que el multiplexado de la capa de transporte necesita (1) que los sockets tengan un identificador único, y (2) que cada segmento tenga campos especiales que indiquen a qué socket ha de entregarse el segmento. Estos campos especiales, son el número de puerto origen y el número de puerto destino (Los segmentos UDP y TCP también tienen otros campos) Cada número de puerto es un número de 16 bits, en un intervalo que va desde 0 a 65535. Los números de puerto del 0 al 1023 se denominan números de puerto bien conocidos, o reservados y están restringidos, ya que son utilizados únicamente por protocolos de aplicación bien conocidos, como HTTP (puerto 80) y FTP (puerto 21).

***Transporte sin conexión: UDP (User Datagram Protocol)***

Como mucho, la capa de transporte tiene que proporcionar un servicio de multiplexado demultiplexado para pasar los datos entre la capa de red y el proceso de aplicación correcto.

UDP, según se definió en RFC 768, hace lo mínimo que un protocolo de transporte tiene que hacer. Aparte de la función de multiplexado/demultiplexado y alguna ligera comprobación de errores, no añade nada a IP. De hecho, si el desarrollador de la aplicación elige UDP en lugar de TCP, entonces la aplicación está hablando casi directamente con IP. UDP toma los mensajes del proceso de aplicación, les añade campos para los números de puerto origen y destino para el servicio de multiplexado demultiplexado, les suma otros dos pequeños campos, y pasa el segmento resultante a la capa de red. La capa de red encapsula el segmento en un datagrama IP. y entonces hace todo lo que puede para entregar el segmento al host receptor. Si el segmento llega al host receptor, UDP utiliza el número de puerto destino para entregar los datos del segmento al proceso de aplicación adecuado. Tenga en cuenta que con UDP no existe un acuerdo entre las entidades de la capa de transporte emisora y las entidades de la capa receptora previo al envío de un segmento. Por este motivo, se dice que UDP es sin conexión.

DNS es un ejemplo de protocolo de la capa de aplicación que normalmente utiliza UDP. Cuando la aplicación DNS en un host quiere realizar una consulta, construye un mensaje de consulta DNS y se lo pasa a UDP. Sin establecer acuerdo alguno con la entidad UDP que se ejecuta en el sistema final destino, UDP añade campos de cabecera al mensaje y se lo pasa a la capa de red. La capa de red encapsula el segmento UDP en un datagrama y se lo envía a un servidor de nombres. La aplicación DNS del host que ha lanzado la consulta espera entonces una respuesta. Si no recibe una respuesta (posiblemente porque la capa de red subyacente perdió la consulta o la respuesta), intentará enviar la consulta a otro servidor de nombres, o informará a la aplicación involucrada de que no puede conseguir una respuesta.

**Por qué elegir UDP en lugar de TCP:**

1. Sin establecimiento de la conexión. TDP utiliza un acuerdo en tres fases antes de comenzar a transferir datos, UDP sólo emite una ráfaga sin ningún formalismo preliminar. De esta forma, no introduce ningún retardo para establecer una conexión. Ésta es probablemente la razón principal de que DNS corra sobre UDP en lugar de sobre TCP (DNS sería mucho más lento si corriera sobre TCP).
2. Sin estado. TCP mantiene el estado de la conexión en los sistemas finales. Este estado de la conexión incluye búferes de envío y recepción, parámetros de control de la congestión, y parámetros de número de secuencia y de reconocimiento. Esta información de estado es necesaria para implementar el servicio de transferencia fiable de datos de TCP y para proporcionar control de congestión. UDP por otro lado, no mantiene estado de conexión alguno, y no realiza seguimiento de estos parámetros. Por esta razón, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de sobre TCP.
3. Poca sobrecarga debida a cabeceras. El segmento TCP tiene una sobrecarga de 20 bytes en cada segmento, mientras que UDP sólo añade ocho bytes.
4. Más control al nivel de aplicación sobre cuándo y qué datos son enviados. Bajo UDP, tan pronto un proceso de aplicación le pasa datos, los empaqueta en un segmento UDP y los pasa a la capa de red. TCP, por su parte, tiene un mecanismo de control de la congestión que ralentiza la capa de transporte emisora cuando uno o más enlaces entre los host origen y destino se encuentran excesivamente congestionados. TCP también continuará enviando un segmento de datos hasta que su recepción haya sido reconocida por el destino, independientemente de cuánto tiempo se emplee en la entrega fiable. Como las aplicaciones de tiempo real a menudo necesitan una tasa de emisión mínima, no quieren retardar la transmisión de segmentos, y pueden tolerar alguna pérdida de datos, el modelo de servicio de TCP no está particularmente preparado para satisfacer las necesidades de estas aplicaciones. Como se ve a continuación, estas aplicaciones pueden utilizar UDP e implementar, como parte de la aplicación, alguna funcionalidad necesaria además del servicio de entrega de segmentos básico entre procesos de UDP.

Las aplicaciones de tiempo real, como la telefonía sobre Internet y la vídeo conferencia, reaccionan mal ante el control de la congestión de TCP. Algunas de estas aplicaciones de tiempo real pueden tolerar perder una pequeña cantidad de paquetes, de forma que una transferencia fiable de datos tampoco resulta crítica para el éxito de las mismas. Aunque hoy es común que las aplicaciones multimedia corran sobre UDP, esto resulta por lo menos controvertido, ya que UDP carece de control de la congestión. Se necesita controlar la congestión para prevenir que la red entre en un estado en el que poco se pueda hacer. Si todo el mundo comenzara a difundir vídeo a tasas altas sin utilizar ningún control de la congestión, habría tal desbordamiento de paquetes en los routers que nadie podría ver nada. Por ello, la falta de control de la congestión en UDP es un problema potencialmente serio.

***Transporte orientado a conexión: TCP (Transmission Control Protocol)***

Se dice que TCP es orientado a conexión porque antes de que un proceso de aplicación pueda comenzar a enviar datos a otro, los dos procesos deben establecer previamente un acuerdo entre ellos (deben enviarse algunos segmentos para establecer los parámetros que aseguren la transferencia de datos). Como parte del establecimiento de la conexión TCP, ambos lados de la conexión iniciarán varias variables de estado TCP.

En la conexión TCP el estado de la misma reside enteramente en los dos sistemas finales. Puesto que el protocolo TCP funciona sólo en los sistemas finales, y no en los elementos intermedios de la red (como routers y bridges), los elementos intermedios de la red no mantienen el estado de la conexión TCP. De hecho, los routers intermedios permanecen completamente ajenos a las conexiones TCP; ellos ven datagramas, no conexiones.

Una conexión TCP proporciona una transferencia de datos full dúplex. Si existe una conexión TCP entre el proceso A en un host y el proceso B en otro host, entonces los datos de la capa de aplicación pueden fluir de A a B al mismo tiempo que los datos de la capa de aplicación Huyen de B a A. Una conexión TCP también es siempre punto a punto, esto es, entre un único emisor y un único receptor. De modo que la multidifusión o multicast (la transferencia de datos de un emisor a varios receptores en una única operación de enviar) no es posible con TCP.

Modo en que se establece una conexión TCP, suponga que un proceso que corre en un host quiere iniciar una conexión con otro proceso en algún otro host. Recuerde que al proceso que inicia la conexión se le denomina proceso cliente, mientras que el otro proceso se llama proceso servidor. El proceso de aplicación cliente primero informa a la capa de transporte del cliente que desea establecer una conexión con un proceso en el servidor. La capa de transporte en el cliente procede entonces a establecer una conexión TCP con el TCP del servidor.

Por ahora, es suficiente con saber que el cliente envía primero un segmento TCP especial: el servidor responde con un segundo segmento TCP especial; y finalmente el cliente responde de nuevo con un tercer segmento especial. Los dos primeros segmentos no contienen carga, esto es, no transportan datos de aplicación; el tercero de estos segmentos puede transportar carga. Debido a que estos tres segmentos son enviados entre los dos host, a este procedimiento de establecimiento a menudo se le suele llamar acuerdo en tres fases.

Una vez que se ha establecido la conexión TCP, los dos procesos de aplicación pueden enviarse datos uno a otro. Consideremos el envío de datos desde el proceso cliente al proceso servidor. El proceso cliente pasa un flujo de datos a través del socket (la puerta del proceso). Una vez que los datos pasan a través de la puerta, los datos están en manos del TCP que se ejecuta en el cliente. TCP dirige estos datos al búfer de envío de la conexión, que es uno de los búferes dispuestos durante el acuerdo en tres tases inicial. De tiempo en tiempo, TCP agarra trozos de datos del búfer de envío para "enviar los datos en segmentos según convenga" (Según la especificación de TCP). La cantidad máxima de datos que puede ser tomada y colocada en un segmento está limitada por el tamaño máximo de segmento (Maximun Segmenl Size MSS). El MSS depende de la implementación de TCP (determinada por el sistema operativo) y a menudo puede ser configurado (Estos tamaños de segmento a menudo se eligen para evitar la fragmentación IP). Tenga en cuenta que el MSS es la máxima cantidad de datos de la capa de aplicación en el segmento, no el tamaño máximo del segmento TCP incluyendo las cabeceras. (Esta terminología es confusa).

TCP asocia cada paquete de datos del cliente con una cabecera TCP, formando así segmentos TCP. Los segmentos se pasan hacia abajo a la capa de red, donde son encapsulados por separado en datagramas IP. Los datagramas IP son enviados entonces por la red. Cuando TCP recibe un segmento en el otro extremo, deposita los datos del segmento en el búfer de recepción de la conexión. La aplicación lee el flujo de datos desde este búfer. Cada lado de la conexión tiene sus propios búferes de envío y recepción. Según se mencionó anteriormente, no se reservan ni búferes ni variables para la conexión en los elementos de la red (routers, bridges y repetidores) entre los host.



***Transferencia fiable de datos***

Recuerde que el servicio de la capa de red de Internet (el servicio IP) no es fiable. IP no garantiza la entrega de los datagramas, no garantiza la entrega ordenada de los mismos, y no garantiza la integridad de los datos que contienen. Con el servicio IP, los datagramas pueden desbordar los almacenes de los routers y nunca alcanzar su destino, pueden llegar en desorden, y sus bits pueden ser corrompidos (intercambiados de 0 a 1 y viceversa). Como los segmentos de la capa de transporte son transportados por la red dentro de datagramas, los segmentos también pueden sufrir estos problemas.

TCP crea un servicio de transferencia fiable de datos sobre el servicio del servicio IP no fiable. El servicio de transferencia fiable de datos de TCP asegura que el flujo de datos que un proceso lee de su búfer de recepción TCP no ha sido corrompido, no tiene agujeros ni duplicación, y está ordenado según la secuencia original; o lo que es lo mismo, que el flujo de bytes es exactamente el mismo flujo de bytes que fue enviado por el sistema final que está al otro extremo de la conexión.

Un emisor TCP, vemos que existen tres eventos principales relacionados con la transmisión y la retransmisión en un emisor TCP: la recepción de datos desde la aplicación que se encuentra por encima; el final del tiempo marcado por el temporizador; y la recepción de un ACK.

Tras la ocurrencia del primero de los eventos principales, TCP recibe datos de la aplicación, los encapsula en un segmento, y pasa dicho segmento a IP. Tenga en cuenta que cada segmento incluye un número de secuencia que es el número del primer byte de datos en el segmento dentro de la secuencia de bytes de la conexión. También debe tener presente que si el temporizador no está ya corriendo por otro segmento. TCP arranca el temporizador cuando se pasa el segmento a IP. (Resulta útil pensar que el temporizador está asociado con el segmento más antiguo por reconocer.)

El segundo evento principal es el evento de fin del tiempo de espera. TCP responde a este evento de fin del tiempo de espera volviendo a transmitir el segmento causante del evento. Entonces. TCP rearranca el temporizador.

El tercer evento principal que debe ser gestionado por el emisor TCP es la llegada de un segmento de reconocimiento (ACK) desde el receptor (más específicamente, de un segmento que contiene un valor válido en el campo ACK). Al ocurrir este evento, TCP compara el valor y del ACK con la variable *BaseEnvio*. La variable de estado *BaseEnvio* de TCP es el número de secuencia del byte más antiguo por reconocer. (Por ello *BaseEnvio-1* es el número del último byte del que se tiene constancia que haya sido recibido correctamente y en orden por el receptor.) Según se indicó anteriormente, TCP utiliza reconocimientos acumulados, de forma que con y se reconoce la recepción de todos los bytes con números anteriores a *y*. Si *y* > *BaseEnvio*, entonces el ACK está reconociendo uno o más segmentos no reconocidos previamente. Por eso, el emisor actualiza su variable *BaseEnvio*, y también rearranca el tiempo si existieran segmentos aún no reconocidos.

*¿Retroceder, o repetición selectiva?*

Analicemos el mecanismo de recuperación de errores considerando la siguiente cuestión: ¿TCP es un protocolo GBN o SR? Recuerde que los reconocimientos TCP son acumulativos, y que los segmentos correctos pero desordenadamente recibidos no son reconocidos individualmente por el receptor. Consecuentemente, el emisor TCP sólo necesita mantener el número de secuencia más pequeño del byte transmitido pero no reconocido (*BaseErvio*), y el número de secuencia para el siguiente byte a enviar (*SígNumSec*). En este sentido, TCP se parece bastante a un protocolo del estilo GBN, pero existen algunas diferencias notables este. Muchas implementaciones TCP almacenarán los segmentos recibidos correctamente, aunque en desorden. Considere también qué sucede cuando el emisor envía una secuencia de segmentos 1, 2, ..., N, y todos los segmentos llegan al receptor en orden y sin errores. Es más, suponga que el reconocimiento para el paquete n < N se pierde pero los N-l reconocimientos llegan al emisor antes del final de sus respectivos tiempos de espera. En este ejemplo, GBN retransmitirá no sólo el paquete n, sino también todos los paquetes que le siguen: n+1, n+2, ..., N. TCP, por su lado, retransmitirá como mucho un segmento, el n, es más, TCP ni siquiera retransmitirá el segmento n si el reconocimiento para el segmento n+1 llegara antes de que se cumpliera el tiempo de espera para el segmento n.

Una modificación propuesta para TCP, denominada reconocimiento selectivo [RFC 2581], permite que un receptor TCP reconozca los segmentos desordenados de forma selectiva, en lugar de reconocer de forma acumulativa el último segmento recibido correctamente y en orden. Cuando se combina con retransmisiones selectivas (saltándose la retransmisión de los segmentos que ya han sido selectivamente reconocidos por el receptor), TCP se parece a nuestro protocolo genérico SR. Así, el mecanismo de recuperación de errores en TCP probablemente se clasifica mejor como un híbrido de los protocolos GBN y repetición selectiva.

***Control de flujo***

Cada lado de una conexión TCP disponen de un almacén para la conexión, cuando la conexión TCP recibe bytes correctos y en orden, coloca los datos en este búfer de recepción. El proceso de aplicación asociado leerá los datos de este búfer, pero no necesariamente en el momento en que lleguen. La aplicación receptora puede estar ocupada con otras tareas, e incluso puede que no intente leer los datos hasta bastante tiempo después de haberlos recibido. Si la aplicación es relativamente lenta en la lectura de los datos, el emisor puede desbordar fácilmente el búfer de recepción de la conexión al enviar muchos datos muy rápidamente.

TCP proporciona un servicio de control del flujo a sus aplicaciones para eliminar la posibilidad de que el emisor desborde el buffer del receptor.

TCP proporciona un control del flujo haciendo que el emisor mantenga una variable denominada ventana de recepción De manera informal, la ventana de recepción se utiliza para dar una idea al emisor de cuánto espacio libre queda en el búfer del receptor. Como TCP es full-dúplex, el emisor que está a cada lado de la conexión mantiene una ventana de recepción distinta. La ventana de recepción, llamada *VentanaRecepcion*, es dinámica.

Existe un problema técnico menor con este esquema. Para verlo, suponga que el búfer de recepción del host B se llena, de forma que *VentanaRecepcion* = 0. Después de indicar al host A que *VentanaRecepcion* = 0, suponga también que B no tiene nada que enviar a A. Considere qué sucede ahora. Cuando el proceso de aplicación en B vacíe el búfer, TCP no enviará segmentos nuevos con el nuevo valor de *VentanaRecepcion* a A: ya que TCP envía un segmento al host A sólo si tiene datos o si tiene un reconocimiento que enviar. Por lo tanto, el host A no es informado de que se ha abierto un espacio en el almacén de recepción de B (¡el host A está bloqueado y no puede transmitir más datos!) Para resolver este problema, la especificación de TCP exige que el host A continúe enviando segmentos con un byte de datos aun cuando la ventana de recepción sea cero. Estos segmentos serán reconocidos por el receptor. Eventualmente, el búfer se irá vaciando, y los reconocimientos contendrán un valor de *VentanaRecepcion* distinto de cero.

Tras describir el servicio de control del flujo, mencionaremos brevemente aquí que UDP no proporciona tal servicio. Para entender esto, considere la emisión de una serie de segmentos UDP desde un proceso en el host A hasta otro proceso en el host B. En una implementación típica de UDP. UDP añade los segmentos en un búfer de tamaño finito que precede al correspondiente socket (la puerta del proceso). El proceso lee un segmento entero del búfer cada vez. Si el proceso no lee los segmentos del búfer lo suficientemente rápido, el búfer se desbordará, y los segmentos serán desechados.

***Aproximaciones al control de la congestión***

En el nivel más amplio, podemos distinguir entre las aproximaciones al control de la congestión teniendo en cuenta la capacidad de la capa de red para proporcionar una asistencia explícita a la capa de transpone con el objetivo de controlar la congestión:

* **Control de la congestión entre extremos**, en un enfoque de extremo a extremo del control de la congestión, la capa de red no proporciona ningún soporte explícito a la capa de transporte para el controlar la congestión. Incluso la misma presencia de congestión en la red debe ser inferida por los sistemas finales basándose exclusivamente en la observación del comportamiento de la red (por ejemplo, pérdidas y retrasos de paquetes). TCP debe adoptar necesariamente este enfoque, ya que la capa IP no proporciona realimentación alguna sobre la congestión de la red a los sistemas finales. La pérdida de los segmentos TCP (ya sea indicada por la finalización del tiempo de espera o por la recepción de tres reconocimientos repetidos) se considera una indicación de la congestión de la red, por lo que TCP, en consecuencia, decrementa su tamaño de ventana. Las nuevas propuestas de TCP utilizan valores incrementales de retardo en el camino de ida y vuelta como indicadores del incremento de la congestión en la red.
* **Control de la congestión asistido por red**, aquí los componentes de la capa de red (los routers) proporcionan una retroalimentación explícita al emisor sobre el estado de la congestión de la red. Esta información puede ser tan simple como un único bit indicando la congestión en un enlace. Este enfoque, que fue adoptado en las primeras arquitecturas de IBM en 1989 y 1990, fue propuesto recientemente para las redes TCP/IP.

En el control de la congestión asistido por red, la información de congestión normalmente se transmite al emisor desde la red seleccionando uno de dos modos disponibles. La realimentación se puede enviar al emisor directamente desde un router. Esta forma de notificación toma normalmente la forma de un paquete obturador que indica la existencia del congestionamiento de la red. La segunda forma de notificación se da cuando un router indica la presencia de congestión al marcar/actualizar un campo en un paquete que fluye desde el emisor al receptor. Cuando el receptor recibe el paquete marcado, notifica al emisor la indicación de congestión. Hay que destacar que esta forma de notificación emplea al menos un intervalo completo de tiempo de ida y vuelta.

***El control de la congestión TCP***

TCP proporciona un servicio de transporte fiable entre dos procesos que corren en host diferentes, otro de sus componentes claves es su mecanismo de control de la congestión, en el que debe utilizar un mecanismo de control de la congestión de extremo a extremo en lugar de uno asistido por la red, ya que la capa IP no proporciona realimentación explícita alguna a los sistemas finales sobre la congestión de la red.

El enfoque tomado por TCP es que cada emisor limita la tasa a la que envía tráfico en su conexión en función de la congestión de la red que percibe. Si un emisor TCP percibe que existe poca congestión en el camino entre él mismo y el destino, incrementa su tasa de emisión; si el emisor percibe que existe congestión a lo largo del camino, reduce su tasa de emisión. Pero este enfoque deja pendientes tres cuestiones:

* Primera, ¿cómo hace el emisor TCP para limitar la tasa de emisión a la que envía tráfico por su conexión?
* Segunda, ¿cómo hace un emisor TCP para percibir que existe congestión en el camino entre él y el destino?
* Y tercera, ¿qué algoritmo debería utilizar el emisor para cambiar su tasa de emisión en función de la congestión entre extremos percibida?

**Supondremos que el emisor TCP está enviando un archivo grande.**

Cada lado de una conexión TCP está compuesto por un búfer de recepción, un búfer de envío, y varias variables. El mecanismo TCP de control de la congestión hace que cada lado de la conexión mantenga una nueva variable: la ventana de congestión. La ventana de congestión, llamada *VentanaCongestion*, impone una restricción sobre la tasa a la que el emisor TCP puede enviar tráfico en la red. Concretamente, la cantidad de datos pendientes de reconocimiento en el emisor no puede exceder del mínimo entre *VentanaCongestion* y *VentanaRecepcion*, esto es:

*UltimoByteLeido* - *UltimoByteReconocido* <= min{*VentanaCongestion*,*VentanaRecepcion*}

Para poder centramos en el control de la congestión (como opuesto al control de flujo), vamos, por lo tanto, a asumir que el búfer de recepción es tan grande que la restricción asociada a la ventana de recepción puede ser ignorada; por ello, la cantidad de datos no reconocidos en el emisor está limitada únicamente por *VentanaCongestion*.

Primero, la anterior restricción limita la cantidad de datos no reconocidos en el emisor, y por lo tanto limita la tasa de envío del emisor. Para ver esto, considere una conexión en la cual las pérdidas de los paquetes y los retardos de transmisión son insignificantes. Entonces, aproximadamente al comienzo de cada tiempo de ida y vuelta (RTT), la anterior restricción permite al emisor enviar *VentanaCongestion* bytes por la conexión, y al final del RTT el emisor recibe los reconocimientos de los datos. Por ello, la tasa de emisión del emisor es más o menos de *VentanaCongestion*/RTT bytes/segundo. Ajustando el valor de *VentanaCongestion*, el emisor puede, por lo tanto, ajustar la tasa a la que envía datos por su conexión.

Consideramos en segundo lugar el modo en que un emisor TCP percibe que existe congestión en el camino entre sí y el destino. Definamos un evento de pérdida en un emisor TCP como la ocurrencia del fin del tiempo de espera, o bien la recepción de tres ACK repetidos. Cuando existe una congestión excesiva, entonces se desbordarán uno (o más) búferes en los routers a lo largo del camino, produciendo datagramas desechados. El datagrama desechado, a su debido tiempo, producirá un evento de pérdida en el emisor (ya sea el fin del tiempo de espera o la recepción de tres ACK repetidos) que será tomado por el emisor como una indicación de la congestión en el camino entre el emisor y el receptor.

Finalmente, en tercer lugar, estamos en posición de considerar el algoritmo que utiliza el emisor TCP para regular su tasa de envío como una función de la congestión percibida. Este algoritmo es el celebrado algoritmo de control de la congestión TCP. El algoritmo tiene tres componentes principales: (i) incremento aditivo, decremento multiplicativo; (ii) arranque lento: y (iii) reacción ante los eventos de fin del tiempo de espera.

*Unidad 4 – Capa de Red*

Los servicios que presta la capa de red son:

Mover paquetes de un host emisor a un host receptor. Tiene 3 funciones:

* **Determinación del camino:** Determinar la ruta o camino tomada por los paquetes. Algoritmos de rutado: algoritmos que calculan dicho camino. La complejidad de los algoritmos de rutado crece según aumenta el número de routers.
* **Encaminamiento:** cuando llega un paquete a la entrada de un router, el router lo moverá hacia el enlace saliente apropiado.
* **Establecimiento de llamada:** la arquitectura de la capa de red requieren que los routers a lo largo del camino desde el emisor al destino, acuerden unos con otros para establecer el estado necesario antes de que los paquetes de datos de la capa de red comiencen a fluir.

La capa de Red permite la conexión desde un host origen a un host destino, enrutando los datagramas IP entre máquinas. En TCP/IP esta capa está implementada en el **protocolo IP (cada componente de Internet que tengan una capa de red deben ejecutarlo)**, e interviene en cada host y encaminador intermedio (router), ya que define los campos del datagrama IP y cómo actúan los sistemas terminales y routers sobre los mismos. El PDU es el datagrama, que encapsula los segmentos de transporte agregándole las direcciones IP origen y destino entre otras cosas.

El protocolo de la capa de transporte (TCP o UDP) en una máquina origen pasa un segmento de la capa de transporte y un destino a la capa IP, del mismo modo que usted proporciona al servicio postal una dirección de destino. La capa IP proporciona entonces el servicio de entrega del segmento a la máquina de destino. Cuando el paquete llega a la máquina destino, IP pasa el segmento a la capa de transporte de la máquina.

***Modelos de Servicios de Red: Datagrama y Circuitos Virtuales***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **¿Todos los paquetes siguen el mismo camino?** | **¿Cuenta con una fase de establecimiento y otra de cierre de circuito?** | **¿Usa mensajes de señalización?** | **¿Usa tablas de enrutamiento?** |
| **Datagrama** | Falso | Falso | Falso | Verdadero |
| **Circuitos Virtuales** | Verdadero | Verdadero | Verdadero | Falso |

El modelo de servicio de red define las características del transpone punto a punto de los datos entre un extremo de la red y el otro; es decir, entre los sistemas finales emisor y receptor.

**¿Datagrama o circuito virtual?**

Puede que la más importante abstracción provista por la capa de red hacia las capas superiores sea el que la capa de red utilice **circuitos virtuales (CV)**. Recordará que una red de paquetes de circuito virtual se porta al estilo de una red telefónica, que emplea circuitos reales, y no circuitos virtuales. Hay tres fases identificables en un circuito virtual:

* **Establecimiento del CV**, durante la fase de inicio, el emisor contacta con la capa de red, especifica la dirección del receptor, y espera a que la red establezca el CV. La capa de red determina el camino entre el emisor y el receptor, esto es, la sucesión de enlaces y conmutadores de paquetes a través de los cuales irán los paquetes del CV. Esto suele implicar la actualización de tablas en cada uno de los conmutadores de paquetes del camino. Durante el inicio del CV, la capa de red deberá también reservar recursos (por ejemplo, ancho de banda) a lo largo del camino del CV.
* **Transferencia de dato**, una vez que el CV ha sido establecido, los datos pueden empezar a fluir por él.
* **Descarte del CV**, se inicia una vez que el emisor (o el receptor) informan a la capa de red de su deseo de terminar el CV. La capa de red informará usualmente al sistema final del otro extremo de la red de la petición de terminación, y actualizará las tablas de cada uno de los conmutadores de paquetes del camino para indicar que el CV deja de existir.

El establecimiento de conexión de la capa de transporte involucra únicamente a los dos sistemas finales. Los dos extremos acuerdan la comunicación y determinan los parámetros de su conexión de capa de transporte antes de comenzar el auténtico flujo de envió de datos en la conexión de la capa de transporte. Aunque ambos sistemas conozcan la conexión de la capa de transporte, los conmutadores de la red son completamente ajenos a él. Por el contrario, con una capa de red de circuito virtual, los conmutadores de paquetes a lo largo del camino entre los dos sistemas finales se involucran en la creación del circuito virtual, y cada conmutador de paquetes está completamente advertido de cada CV que lo atraviesa.

Los mensajes que emiten los sistemas finales hacia la red para indicar el inicio o la terminación de un CV, y los mensajes que se pasan los conmutadores entre ellos para establecer el CV se denominan mensajes de señalización, y los protocolos utilizados para intercambiar estos mensajes se suelen llamar protocolos de señalización.

Con una capa de red de datagrama, cada vez que un sistema final desea enviar un paquete, estampa en el paquete la dirección del sistema final destinatario y lo deposita en la red, esto se hace sin iniciar CV alguno. Los conmutadores de paquetes en una red de datagramas (denominados “routers” en Internet) no guardan información de estado alguna acerca de CV. En lugar de ello, los conmutadores de paquetes encaminan un paquete hacia su destino examinando la dirección destino del paquete, indexando una tabla de encaminamiento con la dirección de destino, y encaminando el paquete en la dirección destino. Puesto que las tablas de encaminamiento pueden modificarse en cualquier momento, una sucesión de paquetes enviados desde un sistema final a otro pueden seguir caminos diferentes a lo largo de la red y llegar en desorden. Internet utiliza una capa de red de datagramas.

Una terminología alternativa al servicio de circuitos virtuales y al de datagrama son el servicio de capa de red orientado a conexión y el servicio de capa de red sin conexión respectivamente. Ciertamente, el servicio de CV es un tipo se servicio orientado a conexión, dado que implica iniciar y descartar una entidad de tipo conexión, y mantener información del estado de la conexión en los conmutadores de paquetes. El servicio de datagrama es un tipo de servicio sin conexión en el que no se emplean entidades de tipo conexión.

La arquitectura actual de Internet sólo proporciona un modelo de servicio, el servicio de datagrama, conocido también como el servicio de mejor esfuerzo, donde no se garantiza en modo alguno la preservación de la temporización entre paquetes; no se garantiza la recepción ordenada según el origen; ni siquiera la entrega final de paquetes.

***Router y Funciones***

El router es considerado como el dispositivo de la capa de red. El mismo se encarga de interconectar redes, es decir, es necesario un router para que un host que se encuentran en diferentes redes puedan comunicarse entre sí.

Sus principales funciones son:

* Determinar el camino de un paquete entre un host y otro.
* Dado un conjunto de routers con enlaces entre ellos, debe encontrar un buen camino (costo mínimo)

***Ruteo en forma estática como dinámica. Funcionamiento e indique ventajas y desventajas***

En los ***algoritmos de ruteo estáticos***, las rutas cambian muy lentamente a lo largo del tiempo, a menudo como resultado de intervención humana (por ejemplo cuando alguien modifica las tablas de ruteo de manera manual). ***Los algoritmos de ruteo dinámicos*** cambian el camino según los cambios en el tráfico de la red o de la topología. Un algoritmo dinámico puede ser ejecutado ya sea de manera periódica o de manera directa con respuesta a los cambios de topología o cambio en los costos de enlace. Mientras que los algoritmos de ruteo dinámicos responden más con los cambios en la red, también son más susceptibles a los problemas tales como oscilaciones en las rutas y lazos en las rutas.

*El ruteo dinámico se divide en estado de enlace y vector de distancia.*

*Estado de enlace* (Dijkstra): la topología de la red y todos los costes de enlaces son conocidos. Esto se puede obtener haciendo que cada nodo realice una transmisión de paquetes estado de enlace hacia todos los demás nodos en la red. Cada paquete contiene las identificaciones y los costes en los enlaces que se encuentran directamente enlazados a ese nodo.

*Vector de distancia*: es iterativo, asincrónico y distribuido. Se dice que es distribuido porque cada nodo recibe información de uno o más de sus nodos vecinos directamente enlazados a él, realiza el cálculo y luego distribuye de regreso el resultado de los cálculos a sus vecinos. Se dice que es iterativo en el sentido de que este proceso continua hasta que no se intercambia información entre los vecinos. Se dice que es asincrónico, pues no requiere que los nodos operen de manera conjunta entre ellos.

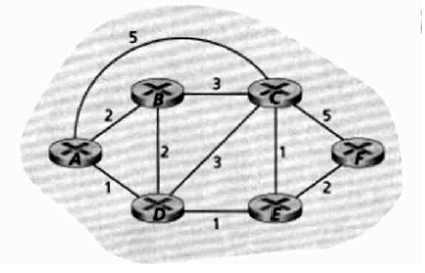
Los algoritmos de ruteo dinámico se dividen en estado enlace y vector distancia. Dado el siguiente cuadro compare:

|  |  |  |  |
| --- | --- | --- | --- |
|  | ¿Cada router conoce la topología de la red completa? | ¿Converge rápidamente? | Protocolos que lo implementan |
| Vector distancia | No | No | RIP, IGRP(Cisco) |
| Estado de enlace ( Vector de caminos ) | Si | Si | OSPF |

Para transferir paquetes de un host fuente a un host destino, la capa de red debe determinar el camino, o ruta, que deberán seguir los paquetes, éste es el trabajo del protocolo de rutado de la capa de red.

Habitualmente, cada host se conecta directamente a un router (router por defecto del host, también conocido como router de primer salto, o router directo, del host). Cuando un host emite un paquete, dicho paquete se transfiere al router por defecto. Dicho router por defecto del host origen se conoce como router fuente, mientras que el router por defecto del host destino será el router destino.

En el núcleo de cualquier protocolo de rutado se halla el algoritmo (algoritmo de rutado) que determina el camino de un paquete desde un router fuente a un router destino. Usualmente, un buen camino es aquél que presenta un coste mínimo (camino más corto). Sin embargo en la práctica, cuestiones del mundo real, como la aplicación de políticas o restricciones entran en juego para complicar los conceptualmente simples y elegantes algoritmos teóricos que subyacen a la práctica del enrutado de las redes actuales.



*El camino de coste mínimo entre los nodos A (fuente) y C (destino) recorre el trayecto ADEC.*

En general, una forma de clasificar los algoritmos de rutado es en función de si son globales o descentralizados:

* Un algoritmo de **rutado global**, calcula el camino de coste mínimo entre una fuente y un destino usando un conocimiento completo (global) de la red. Es decir, el algoritmo toma como entradas la conectividad entre todos los nodos y el coste de los enlaces. Esto precisa que el algoritmo obtenga de alguna forma la información antes de efectuar el cálculo. En la práctica, los algoritmos con información global del estado se conocen como algoritmos de estado de enlaces, dado que el algoritmo debe disponer del coste de cada enlace de la red.

Un algoritmo de estado de enlaces se conoce como algoritmo de Dijkstra. Un algoritmo estrechamente emparentado es el de Prim. El algoritmo de Dijkstra calcula el camino de coste mínimo desde un nodo (la fuente, que denominaremos A) al resto de los nodos de la red.

* En un algoritmo de **rutado descentralizado**, el cálculo del camino de coste mínimo se lleva a cabo de forma iterativa, distribuida. Ningún nodo dispone de la información completa sobre el coste de cada enlace de la red. En lugar de ello, cada nodo comienza conociendo solamente el coste de aquellos enlaces directamente enlazados con él. Después, por un proceso iterativo de cálculo e intercambio de información con sus nodos vecinos (que son aquéllos que se encuentran al otro lado de sus propios enlaces), cada nodo calcula gradualmente el camino de coste mínimo a uno o a un conjunto de destinos. Este tipo de algoritmos es conocido como algoritmo de vector de distancias. Se denomina algoritmo de vector de distancias porque ningún nodo conoce realmente el camino completo desde una fuente a un destino, sino que conoce a qué vecino deberá encaminar cada paquete para alcanzar cierto destino sobre el camino de coste mínimo junto con el coste de dicho camino de coste mínimo.

Mientras que el algoritmo estado de enlaces (EE) es un algoritmo que emplea información global, el algoritmo de vector de distancias (VD), es iterativo, asíncrono, y distribuido. Es distribuido porque cada nodo recibe cierta información de uno o más de sus vecinos directamente enlazados, efectúa cierto cálculo, y puede así distribuir los resultados de su cálculo de vuelta a sus vecinos. Es iterativo porque este proceso continúa hasta que no se intercambia más información entre los vecinos. El algoritmo es asíncrono porque no precisa que todos los nodos operen al unísono uno con otro.

La estructura de datos principal en el algoritmo VD es la tabla de distancias que se mantiene en cada nodo. La tabla de distancias de cada nodo dispone de una fila para cada destino de la red, y de una columna para cada uno de sus vecinos directos. La forma de comunicación vecino a vecino que tendrá lugar en el algoritmo VD: cada nodo debe conocer el coste del camino de coste mínimo de cada uno de sus vecinos a cada destino. Así, cuando un nodo calcula el camino de coste mínimo a algún destino, debe informar a sus vecinos de este nuevo coste mínimo.

Una segunda amplia clasificación de los algoritmos de rutado sigue el criterio de si son estáticos o dinámicos:

* En los algoritmos de **rutado estáticos**, las rutas cambian muy lentamente a lo largo del tiempo, normalmente como resultado de una intervención humana (por ejemplo, porque una persona edita manualmente una tabla de encaminamiento de rutado). Los algoritmos de rutado dinámicos cambian los caminos de rutado según cambia la carga del tráfico o la topología de la red.
* Un algoritmo **dinámico** podrá ejecutarse bien periódicamente, o bien en respuesta directa a un cambio en la topología o en los costes de los enlaces. Si bien los algoritmos dinámicos responden mejor a los cambios en la red, son también más sensibles a problemas como bucles de rutado y oscilaciones en las rutas.

Una tercera forma de clasificar los algoritmos de rutado es en función de si son sensibles a la carga o insensibles a la carga:

* En un algoritmo **sensible a la carga**, el coste de los enlaces varía dinámicamente para reflejar el nivel actual de congestión en el enlace subyacente. Si se asocia un coste elevado a un enlace actualmente congestionado, un algoritmo de rutado intentará escoger rutas que bordeen dicho enlace congestionado. Aunque los algoritmos originales de ARPAnet eran sensibles a la carga, se vieron algunos problemas. Los algoritmos de Internet de hoy en día (como RIP, OSPF, y BGP) son insensibles a la carga, ya que el coste de un enlace no suele reflejar su nivel de congestión actual (o de un pasado reciente).

En Internet suelen usarse tan sólo dos tipos de algoritmos de rutado: un algoritmo de estado de enlaces global dinámico, y un algoritmo de vector de distancias descentralizado dinámico (más utilizado).

***Clase A, Clase B, Clase C - ¿Cuántas hay de cada una? ¿Cuántos hosts pueden haber en cada una?***

Clases A, pocas redes grandes.

Clases B, más redes medianas.

Clases C, muchas redes chicas

* En una red de **clase A**, se asigna el primer octeto para identificar la red, reservando los tres últimos octetos (24 bits) para que sean asignados a los hosts, de modo que la cantidad máxima de hosts es 224 - 2 (se excluyen la dirección reservada para broadcast (últimos octetos en 255) y de red (últimos octetos en 0)), es decir, 16 777 214 hosts.
* En una red de **clase B**, se asignan los dos primeros octetos para identificar la red, reservando los dos octetos finales (16 bits) para que sean asignados a los hosts, de modo que la cantidad máxima de hosts es 216 - 2, o 65 534 hosts.
* En una red de **clase C**, se asignan los tres primeros octetos para identificar la red, reservando el octeto final (8 bits) para que sea asignado a los hosts, de modo que la cantidad máxima de hosts es 28 - 2, ó 254 hosts.

**Direcciones privadas:**

No deberían pasar a la Internet. Filtradas por routers de borde.

- 10.0.0.0 – 10.255.255.255, 1 Clase A.

- 172.16.0.0 – 172.31.255.255, 16 Clases B.

- 192.168.0.0 – 192.168.255.255, 256 Clases C.

***Subredes - Importancia de la máscara de subred***

Cuando una red de computadora es muy grande, conviene dividirla en subredes. Los mismos se usan debido a la escasez de direcciones IP.

**Mascaras de Subred:** es un código numérico que forma parte de la dirección IP de un equipo usado en internet. Afecta solo a un segmento de la red.

**Mascaras por defecto:**

- Para las direcciones de clase A la máscara por defecto es la 255.0.0.0 o /8

- Para las direcciones de clase B la máscara por defecto es la 255.255.0.0 o /16

- Para las direcciones de clase C la máscara por defecto es la 255.255.255.0 o /24

**Dirección reservada:** No se puede asignar la 1era y la última a ningún host. La primera se utiliza como dirección de subred y la última está reservada para broadcast locales dentro de la subred.

***Una máquina que se conecta a Internet, ¿tiene tabla de ruteo?***

Si, Los nodos pueden ser cualquier tipo de dispositivo electrónico conectado a la red. Es la que se ve con el comando 'route' o 'netstat -r'.

***ICMP. Qué hacen los comandos ping y traceroute (tracert en Windows).***

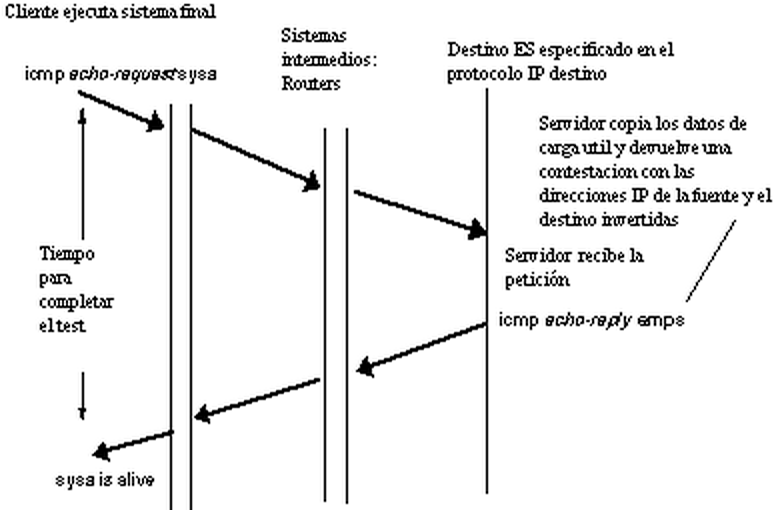
El Protocolo de Mensajes de Control de Internet o ICMP, es el sub protocolo de control y notificación de errores del [Protocolo de Internet](http://es.wikipedia.org/wiki/Protocolo_de_Internet) (IP). Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.

ICMP difiere del propósito de [TCP](http://es.wikipedia.org/wiki/TCP) y [UDP](http://es.wikipedia.org/wiki/User_Datagram_Protocol) ya que generalmente no se utiliza directamente por las aplicaciones de usuario en la red. La única excepción es la herramienta [ping](http://es.wikipedia.org/wiki/Ping) y [traceroute](http://es.wikipedia.org/wiki/Traceroute), que envían mensajes de petición Echo ICMP (y recibe mensajes de respuesta Echo) para determinar si un host está disponible, el tiempo que le toma a los paquetes en ir y regresar a ese host y cantidad de hosts por los que pasa.

El comando Ping nos indica el tiempo exacto que tardan los paquetes de datos en ir y volver a través de la red desde nuestra PC a un determinado servidor remoto.

Se suele utilizar digitando en la línea de comandos ping + IP\_del\_otro\_PC

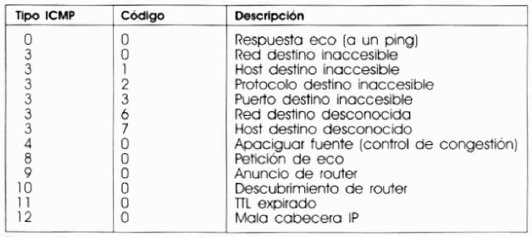
**Tracerout**: Se utiliza para analizar el camino que un paquete toma para alcanzar su destino



A continuación discutiremos el protocolo de mensajes de control de Internet (ICMP; internet Control Message Protocol), que se emplea en host, routers y pasarelas para comunicar información de la capa de red de uno a otro, ICMP viene especificado en RFC 792. El uso más común de ICMP es para informar de errores (aunque no son su único uso). En algún punto, un router IP fue incapaz de encontrar un camino al host especificado en su cliente Telnet, FTP o HTTP. El router creó y envió un mensaje ICMP de tipo 3 a su host indicando el error. Su host recibió el mensaje ICMP y devolvió el código de error al software TCP, que intentó conectarse al host remoto. TCP, a su vez, devolvió el código de error a su aplicación.

ICMP suele considerarse parte de IP, pero arquitectónicamente se halla justo sobre IP, dado que los mensajes ICMP van en el interior de paquetes IP. Esto es, los mensajes ICMP viajan sobre IP, igual que los segmentos TCP o UDP viajan sobre IP. De igual modo, cuando un host recibe un paquete IP donde se especifica ICMP como protocolo de capa superior, éste demultiplexa el paquete a ICMP, de la misma forma que demultiplexa un paquete a TCP o a UDP.

Los mensajes ICMP tienen un campo de tipo y de código, y también contienen los primeros ocho bytes del datagrama IP que provocó el envío del mensaje ICMP en primer lugar (de forma que el emisor pueda determinar qué paquete causó el error).



Los mensajes ICMP no sólo se usan para señalar condiciones de error. El conocido programa "ping" envía un mensaje ICMP de tipo 8 y código 0 al host especificado. El host destino, al ver la petición de eco, devuelve una respuesta de eco ICMP de tipo 0 y código 0. Otro mensaje ICMP interesante es el mensaje de apaciguamiento de fuente (source quench). Este mensaje es muy poco frecuente. Su propósito original era efectuar un control de congestión, para permitir que un router congestionado enviara un mensaje de apaciguamiento de fuente a un host para forzar a éste a que redujera su tasa de transmisión. TCP tiene su propio mecanismo de control de la congestión que opera en la capa de transporte, sin usar la realimentación de la capa de red como el mensaje ICMP de apaciguamiento de la fuente.

El programa "Traceroute", que nos permite seguir la ruta de un host a cualquier otro host del mundo. Curiosamente, está implementado con mensajes ICMP. Para determinar los nombres y direcciones de los routers entre la fuente y el destino, Traceroute en origen envía una serie de datagramas IP corrientes al destino. El primero de ellos tiene un TTL de 1, el segundo de 2, el tercero de 3, y así sucesivamente. La fuente también inicia temporizadores para cada uno de los datagramas. Cuando el n-ésimo datagrama llega al n-ésimo router, el n-ésimo router observa que el TTL del datagrama acaba de expirar. Según las reglas del protocolo IP, el router desecha el datagrama y envía una advertencia ICMP a la fuente (tipo 11 y código 0). Este mensaje de advertencia incluye el nombre del router y su dirección IP. Cuando este mensaje ICMP llega a la fuente, ésta obtiene el tiempo de ida y vuelta a partir del temporizador y el nombre y la dirección IP del n-ésimo router a partir del mensaje ICMP.

***Dirección 127.0.0.1***

La dirección 127.0.0.1, que también significa "este mismo dispositivo", sí se puede usar como dirección de destino y el efecto es que los mensajes que se le envíen "rebotan" y vuelven a ser recibidos por el mismo dispositivo. Esto es muy útil para propósitos de pruebas. Por ejemplo, podemos correr un servidor Web en la misma computadora donde estemos haciendo páginas .html dinámicas; cuando queramos ver cómo lucen, podemos correr el navegador con el URL "http://127.0.0.1/index.html". Esta dirección también se llama "loopback address".

Es lo mismo que localHost. Es nuestra propia PC

***NAT y para qué sirve - Ejemplo de su uso y analice cómo funcionaría en ese entorno.***

**NAT** (Network Address Translation - Traducción de Dirección de Red) es un mecanismo utilizado por [routers](http://es.wikipedia.org/wiki/Enrutador) IP para intercambiar paquetes entre dos redes que se asignan mutuamente [direcciones](http://es.wikipedia.org/wiki/Dirección_IP) incompatibles. Consiste en convertir en tiempo real las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes para permitir la operación de [protocolos](http://es.wikipedia.org/wiki/Protocolo) que incluyen información de direcciones dentro de la conversación del protocolo.

Su uso más común es permitir utilizar direcciones privadas (definidas en el [RFC 1918](http://tools.ietf.org/html/rfc1918)) para acceder a [Internet](http://es.wikipedia.org/wiki/Internet). Existen rangos de [direcciones privadas](http://es.wikipedia.org/wiki/Red_privada) que pueden usarse libremente y en la cantidad que se quiera dentro de una red privada. Si el número de direcciones privadas es muy grande puede usarse solo una parte de direcciones públicas para salir a Internet desde la red privada. De esta manera simultáneamente sólo pueden salir a [Internet](http://es.wikipedia.org/wiki/Internet) con una dirección [IP](http://es.wikipedia.org/wiki/Dirección_IP) tantos equipos como direcciones públicas se hayan contratado. Esto es necesario debido al progresivo [agotamiento de las direcciones IPv4](http://es.wikipedia.org/wiki/Agotamiento_de_las_direcciones_IPv4). Se espera que con el advenimiento de [IPv6](http://es.wikipedia.org/wiki/IPv6) no sea necesario continuar con esta práctica.

El protocolo [TCP/IP](http://es.wikipedia.org/wiki/TCP/IP) tiene la capacidad de generar varias conexiones simultáneas con un dispositivo remoto. Para realizar esto, dentro de la cabecera de un paquete IP, existen campos en los que se indica la dirección origen y destino. Esta combinación de números define una única conexión.

Una pasarela NAT cambia la dirección origen en cada paquete de salida y, dependiendo del método, también el puerto origen para que sea único. Estas traducciones de dirección se almacenan en una tabla, para recordar qué dirección y puerto le corresponde a cada dispositivo cliente y así saber donde deben regresar los paquetes de respuesta. Si un paquete que intenta ingresar a la red interna no existe en la tabla de en un determinado puerto y dirección se pueda acceder a un determinado dispositivo, como por ejemplo un servidor web, lo que se denomina NAT inverso o DNAT (Destination NAT).

NAT tiene muchas formas de funcionamiento, entre las que destacan:

**Estática**

Es un tipo de NAT en el que una dirección IP privada se traduce a una dirección IP pública, y donde esa dirección pública es siempre la misma. Esto le permite a un [host](http://es.wikipedia.org/wiki/Host), como un servidor Web, el tener una dirección IP de red privada pero aun así ser visible en Internet.

**Dinámica**

Es un tipo de NAT en la que una dirección IP privada se mapea a una IP pública basándose en una tabla de direcciones de IP registradas (públicas). Normalmente, el router NAT en una red mantendrá una tabla de direcciones IP registradas, y cuando una IP privada requiera acceso a Internet, el router elegirá una dirección IP de la tabla que no esté siendo usada por otra IP privada. Esto permite aumentar la seguridad de una red dado que enmascara la configuración interna de una red privada, lo que dificulta a los hosts externos de la red el poder ingresar a ésta. Para este método se requiere que todos los hosts de la red privada que deseen conectarse a la red pública posean al menos una IP pública asociadas.

**Sobrecarga**

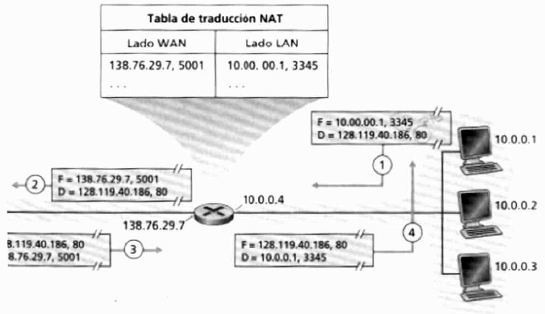
La más utilizada es la NAT dinámica, conocida también como [PAT](http://es.wikipedia.org/wiki/Port_address_translation) (Port Address Translation -Traducción de Direcciones por Puerto), NAT de única dirección o NAT multiplexado a nivel de puerto.

**Solapamiento**

Cuando las direcciones IP utilizadas en la red privada son direcciones IP públicas en uso en otra red, el ruteador posee una tabla de traducciones en donde se especifica el reemplazo de éstas con una única dirección IP pública. Así se evitan los conflictos de direcciones entre las distintas redes.

En un hogar con un router y 3 maquinas conectadas a él, las mismas tienen la misma IP pública (en internet) que es la que asigna el ISP local; pero “adentro” de la LAN tienen IPs privadas (del rango reservado para direcciones privadas de cada clase A, B, C o D). Entonces el router hace NAT, esto es editar cada paquete emitido por alguna maquina, cambiando la IP origen (que viene de la LAN y es privada) por la IP publica, también puede hacer PAT (traducción de puertos).

Existe una aproximación más simple a la asignación de direcciones que ya ha encontrado una amplia difusión en estos escenarios: la traducción de direcciones de red (NAT; network address translstion) [RFC 2663; RFC 3022].



Un router habilitado para NAT, que reside en casa, tiene una interfaz, que es parte de la red casera (en la parte derecha de la imagen). El direccionamiento en la red de la casa es exactamente igual a como ya se ha visto: las cuatro interfaces de la red de la casa tienen la misma dirección de red 10.0.0/24. La diferencia con nuestros casos anteriores se encuentra en la relación entre el router de la casa y el ISP. El router habilitado para NAT no ejecuta un protocolo de rutado inter-SA con el router correspondiente del ISP; en realidad, el router habilitado para NAT ni siquiera parece un router desde el exterior, sino que se comporta hacia fuera como un único dispositivo con una única dirección IP. Por ello, todo el tráfico saliente del router de la casa hacia Internet tiene como dirección fuente IP 138.76.29.7, y todo el tráfico entrante en la casa debe ir destinado a la dirección 138.76.29.7. Esencialmente, el router habilitado para NAT está ocultando los detalles de la red de la casa al mundo exterior. (A parte de esto, usted puede que se esté preguntando cómo consiguen sus direcciones IP el resto de los componentes de la casa, y cómo consigue el router su dirección IP. La respuesta es, por supuesto, la misma: de DHCP. El router obtiene su dirección del servidor DHCP del ISP, y el router dispone de un servidor DHCP para dar direcciones a los computadores del espacio de direcciones de red controlado por el router DHCP-NAT de la casa.)

Ahora bien: si todos los datagramas que llegan al router NAT desde la WAN tienen la misma dirección IP de destino (concretamente la de la interfaz WAN del router NAT), se preguntará cómo es capaz el router de distinguir a quién debe enviar cada datagrama concreto. El truco se encuentra en el empleo de una tabla de traducción NAT ubicada en el router NAT que contiene entradas con números de puerto, así como direcciones IP.

***RFC 1918 y cómo se relaciona con NAT?***

Especifica direcciones Privadas

* No tienen significado global no son únicas.
* Definidas en RFC-1918.
* Se utilizan en Intranets. Redes autónomas sin conexión a Internet.
* Para conectar a Internet requieren un proceso de transformación: NAT, RFC-1631.
* No deberían pasar a la Internet. Filtradas por routers de borde.
  + 10.0.0.0 – 10.255.255.255, 1 Clase A.
  + 172.16.0.0 – 172.31.255.255, 16 Clases B.
  + 192.168.0.0 – 192.168.255.255, 256 Clases C.

Si un dispositivo de una red privada necesita comunicarse con otro dispositivo de otra red privada distinta, es necesario que cada red cuente con una [puerta de enlace](http://es.wikipedia.org/wiki/Puerta_de_enlace) con una dirección IP pública, de manera que pueda ser alcanzada desde fuera de la red y así se pueda establecer una comunicación, ya que un enrutador podrá tener acceso a esta puerta de enlace hacia la red privada. Típicamente, esta puerta de enlace será un dispositivo de [traducción de dirección de red](http://es.wikipedia.org/wiki/Traducci%C3%B3n_de_direcci%C3%B3n_de_red) (NAT) o un [servidor proxy](http://es.wikipedia.org/wiki/Proxy).

El documento [RFC 1597](http://tools.ietf.org/html/rfc1597) contiene la especificación original y permanece por razones históricas, pues ha sido reemplazado por el documento [RFC 1918](http://tools.ietf.org/html/rfc1918).

Esto se relaciona con NAT dándole uso a las direcciones privadas en una red local y atreves de NAT traducirlas a IP públicas.

***Ipv6 - Enumere diferencias existentes en el formato de datagramas respecto de IPv4.***

**Problemas con IPv4**

* Direcciones IPv4 no disponibles.
* Tablas de ruteo muy grandes en el backbone de Internet.
* Carencia de seguridad a nivel IP.
* Deficiente manejo de la Calidad de Servicio (QoS).
* Inexistencia de una fácil auto-configuración y remuneración de direcciones.
* Problemas con la movilidad.

**Beneficios de IPv6**

* Mayor espacio de direcciones (128 bits):
* 340.282.366.920.938.463.463.374.607.431.768.211.456 dirs.
* Formato de cabecera simplificado.
* Auto-configuración de direcciones (plug and play).
* Arquitectura de red jerárquica para un ruteo eficiente.
* Seguridad a nivel IP (IPSec obligatorio).
* Menor overhead de procesamiento.
* Jumbogramas, size(datagrama) > 64KB.
* Movilidad y más direcciones de multicast.

El principal motivo del esfuerzo fue el hecho de que el espacio de direcciones IP de 32 bits estaba comenzando a agotarse, al añadir a marchas forzadas redes y nodos IP (para cada uno de las cuales se precisaba de una dirección única). Para responder a esta necesidad de un gran espacio de direcciones IP, se desarrolló un nuevo protocolo IP: IPv6.

El protocolo Internet versión 6 (IPv6) es una nueva versión de IP (Internet Protocol), definida en el RFC 2460 y diseñada para reemplazar a la versión 4 (IPv4) RFC 791, actualmente en uso dominante.  
IPv4 posibilita 4.294.967.296 (232) direcciones de red diferentes, un número inadecuado para dar una dirección a cada persona del planeta, y mucho menos a cada vehículo, teléfono, PDA, etcétera. En cambio, IPv6 admite 340.282.366.920.938.463.463.374.607.431.768.211.456 (2128 o 340 sixtillones de direcciones)  
 En muchos aspectos, IPv6 es una extensión conservadora de IPv4. La mayoría de los protocolos de transporte -y aplicación- necesitan pocos o ningún cambio para operar sobre IPv6; las excepciones son los protocolos de aplicación que integran direcciones de capa de red, como FTP o NTPv3, NTPv4.

IPv6 especifica un nuevo formato de paquete, diseñado para minimizar el procesamiento del encabezado de paquetes. Debido a que las cabeceras de los paquetes IPv4 e IPv6 son significativamente distintas, los dos protocolos no son interoperables.

Cambios:

* El campo con la longitud de la cabecera se ha eliminado en IPv6 debido a que la longitud de la cabecera es fija en esta versión.
* La suma de comprobación (checksum) de la cabecera se ha eliminado debido a que la suma de comprobación es ofrecida por los protocolos de nivel superior.

-------------------------------------------------------------------------------------------

* El campo con el tipo de servicio y la notificación explicita de congestión se ha eliminado en IPv6. El campo con la prioridad y la etiqueta de flujo juntas toman la misma función.
* La longitud total se ha eliminado en IPv6 y se ha reemplazado por la longitud de carga o de datos.
* La identificación, los indicadores (flags) y el desplazamiento (fragment offser) se ha eliminado de la cabecera base en IPv6 y se han incluido en la cabecera de aplicación.
* El campo TTL (time to live) se denomina en IPv6 límite de saltos (hop limit).
* El campo protocolo se ha reemplazado por el campo cabecera siguiente.
* El campo opciones en IPv4 se implementa como cabeceras de ampliación en IPv6.

Tenga en cuenta que un datagrama IPv4 tiene un total de 20 bytes de cabecera (suponiendo que no hubiera opciones). Si el datagrama lleva un segmento TCP, cada datagrama (sin fragmentar) lleva un total de 40 bytes de cabecera (20 bytes de cabecera IP y 20 bytes de cabecera TCP), aparte del mensaje de la capa de aplicación.

IPv6 ha introducido un nuevo tipo de direcciones, denominadas direcciones anycast (difusión hacia algún enlace), que permiten dirigir un datagrama a una dirección anycast que será entregada a cualquier host del grupo denotado por la dirección.

Se han eliminado o convertido en opcionales ciertos campos de IPv4. La cabecera resultante, de longitud fija de 40 bytes, permite un procesamiento más rápido del datagrama IP.

IPv6 incorpora una etiqueta de flujo. Aunque este protocolo tiene una definición un tanto elusiva de flujo [RFC 1752 y 2460], sostienen que esto permite "etiquetar los paquetes pertenecientes a ciertos flujos para los cuales el emisor espera un tratamiento especial, como una calidad de servicio mejorada o un servicio de tiempo real". Por ejemplo, la transmisión de sonido y video sería tratada mejor como un flujo. Sin embargo, ciertas aplicaciones más tradicionales, la transferencia de archivos y el correo electrónico, no tendrían por qué tratarse como flujos. Es posible que el tráfico transmitido por un usuario de alta prioridad (por ejemplo, alguien que pagara por un mejor servicio para su tráfico) pudiera ser tratado también como un flujo. Parece claro, de cualquier modo, que los diseñadores de IPv6 prevén la necesidad eventual de diferenciar entre los flujos, incluso aunque el significado exacto de tal característica no haya sido determinado aún. La cabecera IPv6 también tiene un campo de clase de tráfico de 8 bits. Este campo, como el campo TOS (Type of Service) de IPv4, puede usarse para priorizar los datagramas de ciertas aplicaciones (por ejemplo los paquetes ICMP) sobre los datagramas de otras aplicaciones (por ejemplo la red de noticias).

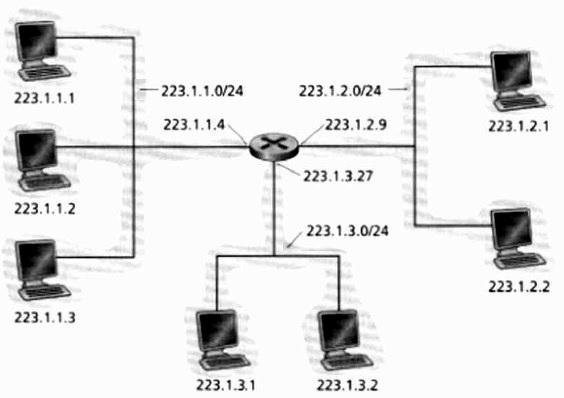
Un host suele tener un solo y punto de enlace con la red. Cuando el IP de un host precisa enviar un datagrama, lo hace sobre este enlace. La frontera entre el host y el enlace físico se denomina interfaz. Por otra parte, un router es esencialmente diferente de un host. Dado que el ti abajo del router es justamente recibir un datagrama de un enlace entrante y encaminar dicho datagrama hacia algún enlace saliente, es preciso que disponga de más de un enlace con la red. La frontera entre el router y cualquiera de sus enlaces se denomina también interfaz. Por ello, un rou.er tendrá varias interfaces, una por cada enlace. Puesto que cada host y cada router son capaces de enviar y recibir datagramas IP. IP precisa que cada interfaz de cada host y de cada router se distingan por su propia dirección IP. Técnicamente hablando, la dirección IP es una característica asociada a la interfaz de cada host o router, más que a la máquina a la que sirve.

Cada dirección IP tiene un tamaño de 32 bits (o cuatro bytes), con lo que hay un total de 232 direcciones IP posibles. Pongamos por caso la dirección IP 193.32.216.9. El 193 es el equivalente decimal de los primeros ocho bits de la dirección; el 32 es el equivalente decimal de los siguientes ocho bits, y así sucesivamente. Así, la dirección 193.32.216.9 en notación binaria es:

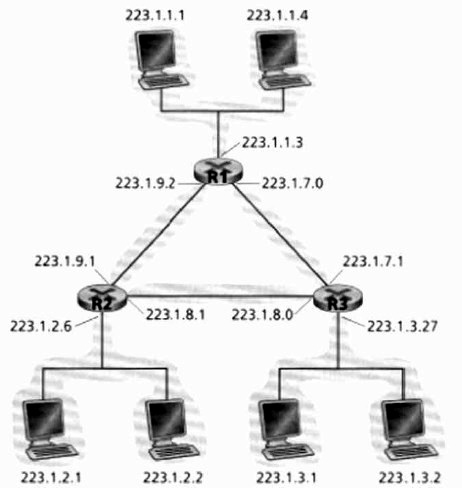
11000001 00100000 11011000 00001001

Cada interfaz de cada host o router de toda Internet debe tener una dirección IP globalmente única.

Ejemplo de direccionamiento e interfaces IP. En esta figura se muestra un router (con tres interfaces) dedicado a conectar siete host. Preste atención a las direcciones IP asignadas a las interfaces de los host y del router, y observe ciertos detalles. Los tres host de la parte superior izquierda de la figura, y la interfaz del router a la que se conectan, tienen todos una dirección IP de la forma 233.1.1.xxx. Esto es, comparten los primeros 24 bits de su dirección IP. Además están conectados entre sí por un único enlace físico, sin routers entre medias. En la jerga de IP, las interfaces de estos tres host y la interfaz superior izquierda del router forman una red IP, o simplemente una red. Los 24 bits de dirección comunes forman la parte de red de su dirección IP; los ocho bits restantes son la parte de host de la dirección IP. (Preferiríamos usar la expresión “parte de interfaz de la dirección" en lugar de “parte de host de la dirección", dado que una dirección IP es en realidad cuestión de la interfaz más que del host: pero en la práctica se emplea la expresión “parte de host".) La red en si también tiene una dirección: 223.1.1.0 24, donde la notación “/24”, también conocida como máscara de red, indica que la dirección de la red viene determinada por los primeros 24 bits de los 32 de que consta la dirección de red. Estos primeros bits que definen la dirección de red también se denominan prefijo de red. En definitiva, la red 223.1.1.0 24 consta de tres interfaces de host (223.1.1.1, 223.1.1.2, y 223.1.1.3) más una interfaz (223.1.1.4) del router. Cualquier host añadido con posterioridad a la red 223.1.1.0 24 precisará de una dirección de la forma 223.1.1.xxx. En la Figura 4.14 se muestran dos redes más: la red 223.1.2.0 24 y la red 223.1.3.0



La definición de red IP no se restringe a segmentos Ethernet que conecten diversos host con la interfaz de un router. En la siguiente imagen, que muestra tres routers interconectados por enlaces punto a punto. Cada router tiene tres interfaces, una para cada enlace punto a punto, y una para cada enlace de difusión que conecta dicho host directamente con su par de host. ¿Qué redes IP hay aquí? Tres de ellas (223.1.1.0 24, 223.1.2.0/24. y 223.1.3.0 24) tienen una finalidad similar a las redes que mostraba la anterior. Pero dése cuenta de que, además, hay otras tres redes: una red (223.1.9.0/24) para las interfaces que conectan los routers R 1 y R2: otra red (223.1.8.0/24) para las interfaces que conectan los routers R2 y R3; y una tercera red (223.1.7.0/24) para las interfaces que conectan los routers R3 y R1.



La Internet real consta de millones de redes como ésta. La noción de red y de dirección de red es importante, y juega un papel central en la arquitectura de rutado de Internet.

***CIDR (Class Interdomain routing) ¿Por qué resulta útil?***

Encaminamiento Inter – Dominios sin clase, representa la última mejora en el modo como se interpretan las direcciones IP. Su introducción permite mayor flexibilidad al dividir rangos de direcciones IP en redes separadas. De esta manera permite:

* Un uso más eficiente de loas cada vez más escasas direcciones IPV4
* Un mayor uso de jerarquías de direcciones (“agregación de prefijos de red”), disminuyendo la sobrecarga de los enrutamientos principales de internet para realizar el encaminamiento.

CIDR reemplaza la sintaxis previa para nombrar direcciones IP, las clases de redes. En vez de asignar bloques de direcciones en los límites de los octetos, que implicaban prefijos naturales de 8, 16 y 24 bits, CIDR usa la técnica [VLSM](http://es.wikipedia.org/wiki/VLSM) (Variable-Length Subnet Masking - Máscara de Subred de Longitud Variable), para hacer posible la asignación de prefijos de longitud arbitraria.

CIDR engloba:

* La técnica [VLSM](http://es.wikipedia.org/wiki/VLSM) para especificar prefijos de red de longitud variable. Una dirección CIDR se escribe con un sufijo que indica el número de bits de longitud de prefijo, p.ej. 192.168.0.0/16 que indica que la máscara de red tiene 16 bits (es decir, los primeros 16 bits de la máscara son 1 y el resto 0). Esto permite un uso más eficiente del cada vez más escaso espacio de direcciones IPv4
* La agregación de múltiples prefijos contiguos en superredes, reduciendo el número de entradas en las tablas de ruta globales.

Esta técnica es muy eficiente en el uso y manejo de redes de área local.

***Sistemas autónomos - ¿Qué es necesario para que los distintos sistemas autónomos puedan rutear el tráfico de hosts pertenecientes a diferentes sistemas autónomos?***

Un sistema autónomo (SA): es un conjunto de redes y dispositivos router IP bajo la misma administración y utilizando un protocolo de pasarela interno (IGP) o combinación de IGP para rutear internamente, independientemente de la red de su proveedor. Cada SA en internet debe tener un número identificador, Solucionan los problemas de escala y de autoridad administrativa (Algoritmo de rutado).

Todos los routers que están dentro del mismo SA ejecutan el mismo algoritmo de rutado y tienen información una de otro. El algoritmo de rutado que se ejecuta dentro de un SA se denomina protocolos de rutado de Sistemas intra - autónomo. Estos routers se conocen como routers pasarelas. Para que los routers pasarelas ruten paquetes de un sistema a otro, las pasarelas deben saber cómo determinar los caminos de rutado entre ellos. El algoritmo de rutado que emplean las pasarelas para rutar entre varios SA se conocen como protocolos de rutado de sistemas inter-autónomo.

Técnicamente un Sistema Autónomo se define como “un grupo de redes IP que poseen una política de rutas propia e independiente”. Esta definición hace referencia a la característica fundamental de un Sistema Autónomo: realiza su propia gestión del tráfico que fluye entre él y los restantes Sistemas Autónomos que forman Internet. Aun considerando que el ISP podía soportar múltiples sistemas autónomos, Internet solo considera la política de definición de trayectorias establecida por el ISP. Por lo tanto, el ISP debería contar con un ASN registrado. Un número de AS o ASN se asigna a cada AS para ser utilizado por el esquema de encaminamiento BGP, este número identifica de manera única a cada red dentro del Internet.  
Los protocolos de ruteo que utilizan internamente se denominan IGP (Interior Gateway Protocol), y pueden ser: RIP, IGRP, EIGRP, OSPF, IS-IS...  
Los SA permiten el ruteo jerárquico, tal y como la mayor parte del encaminamiento en [TCP/IP](http://es.wikipedia.org/wiki/TCP/IP) se basa en la jerarquía de encaminamientos de dos niveles, según la cual una [dirección IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) se divide en una parte para la [red](http://es.wikipedia.org/wiki/Red) y otra para el [servidor](http://es.wikipedia.org/wiki/Servidor). Los gateways usan únicamente la parte de la red hasta que una [datagrama](http://es.wikipedia.org/wiki/Datagrama) IP llega a un gateway que puede entregar el paquete directamente al host. Se pueden añadir niveles adicionales de encaminamiento utilizando subredes.

Cada SA se conecta a un router de borde o gateway, que lo conecta a otros. Se denominan EGP (Exterior Gateway Protocols) a los protocolos para encaminar información entre distintos AS (GGP, EGP, BGP), y que a diferencia de los que son IGP, éste no usa métricas (numero de saltos, ancho de banda, retardo) ya que toma decisiones de enrutado basándose en políticas de la red.  
El ruteo jerárquico permite salvar los problemas que supondrían utilizar los mismos protocolos entre diferentes subredes:

* Escala: El enorme tamaño que alcanzarían las tablas de ruteo, el overhead de intercambiar información entre los routers para una red de grandes dimensiones y el enorme tiempo de convergencia que se tendría.
* Autonomía administrativa: La imposibilidad de elegir un protocolo específico por parte de los administradores de las redes.

Internal Gateway Protocol (IGP) es un protocolo de pasarela interno, hace referencia a los protocolos usados dentro de un [sistema autónomo](http://es.wikipedia.org/wiki/Sistema_aut%C3%B3nomo). Por otra parte, un [Protocolo de Pasarela Externo](http://es.wikipedia.org/wiki/Exterior_Gateway_Protocol) (EGP) determina si la red es accesible desde el [sistema autónomo](http://es.wikipedia.org/wiki/Sistema_aut%C3%B3nomo), y usa el IGP para resolver el encaminamiento dentro del propio sistema. El protocolo más utilizado de EGP es BGP

Los protocolos de pasarela internos se pueden dividir en dos categorías:

* Protocolo de enrutamiento vector-distancia
  + RIPv1, RIPv2
  + IGRP
  + GGP
* Protocolo de enrutamiento vector-camino
  + BGP
  + EGP
* Protocolo de enrutamiento enlace-estado
  + OSPF
  + IS-IS
* Protocolo de enrutamiento hibrido
  + EIGRP

|  |  |  |  |
| --- | --- | --- | --- |
|  | ¿Implementaciones más conocidas? | ¿Dónde se usan? | ¿Pueden ser sustituidos por configuración manual (Si, Depende, No)? ¿Por qué? |
| IGP | RIP (Rputing Internet Protocol), IGRP (Interior Gateway Routing Protocol), EIGRP, OSPF (Open Shortest Path First), IS-IS (Intermediate System to Intermediate System) | Trabajar en un mismo sistema autónomo |  |
| EGP | GGP (Gateway to Gateway protocol), EGP (Exterior Gateway Protocol), BGP (Border Gateway Protocol) | Trabajar entre sistemas autónomos |  |

***DHCP***

DHCP (protocolo de configuración dinámica de [host](http://es.wikipedia.org/wiki/Host)) es un [protocolo de red](http://es.wikipedia.org/wiki/Protocolo_de_red) que permite a los clientes de una red [IP](http://es.wikipedia.org/wiki/Protocolo_de_Internet) obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo [cliente/servidor](http://es.wikipedia.org/wiki/Cliente/servidor) en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después. Este protocolo se publicó en octubre de 1993, y su implementación actual está en la [RFC 2131](http://tools.ietf.org/html/rfc2131). Para DHCPv6 se publica el [RFC 3315](http://tools.ietf.org/html/rfc3315).

Antiguamente cada dirección IP debía configurarse manualmente en cada dispositivo y, si el dispositivo se mueve a otra subred, se debe configurar otra dirección IP diferente. El DHCP le permite al administrador supervisar y distribuir de forma centralizada las direcciones IP necesarias y, automáticamente, asignar y enviar una nueva IP si fuera el caso en el que un dispositivo es conectado en un lugar diferente de la red.

El protocolo DHCP incluye tres métodos de asignación de direcciones IP:

* Asignación manual o estática: Asigna una dirección IP a una máquina determinada. Se suele utilizar cuando se quiere controlar la asignación de dirección IP a cada cliente, y evitar, también, que se conecten clientes no identificados.
* Asignación automática: Asigna una dirección IP de forma permanente a una máquina cliente la primera vez que hace la solicitud al servidor DHCP y hasta que el cliente la libera. Se suele utilizar cuando el número de clientes no varía demasiado.
* Asignación dinámica: el único método que permite la reutilización dinámica de las direcciones IP. El administrador de la red determina un rango de direcciones IP y cada dispositivo conectado a la red está configurado para solicitar su dirección IP al servidor cuando la tarjeta de interfaz de red se inicializa. El procedimiento usa un concepto muy simple en un intervalo de tiempo controlable. Esto facilita la instalación de nuevas máquinas clientes a la red.

Algunas implementaciones de DHCP pueden actualizar el [DNS](http://es.wikipedia.org/wiki/DNS) asociado con los servidores para reflejar las nuevas direcciones IP mediante el protocolo de actualización de DNS establecido en [RFC 2136 (Inglés)](http://www.faqs.org/rfcs/rfc2136.html).

Funcionamiento de DHCP:

* Los host al arrancar solo tienen acceso a su red local de forma broadcast.
* En la red local existe un o más servidor de auto-configuración (DHCP servers).
* Los host sin parámetros de red envían requerimiento.
* Los servidores los atienden asignando los valores que brindan conectividad.
* El parámetro se reserva por un tiempo.

Algunos Mensajes DHCP son “Discover”, “Offer”, “Request”, “ACK”, “Release”, “NAK” y están montados sobre UDP en el puerto 68 (client) y el puerto 67 (server).

***Obtención de direcciones de host***

A partir del momento en que una organización ha obtenido un bloque de direcciones de su ISP, puede asignar direcciones IP individuales a cada interfaz de host y router de su organización. Para las direcciones de interfaces de routers, el administrador del sistema configura estas direcciones IP manualmente en cada router (frecuentemente de modo remoto, con una herramienta de gestión). Hay dos formas de distribuir una dirección IP a cada host.

* Manualmente, un administrador del sistema configura manualmente la dirección IP de cada host (usualmente en un archivo).
* Por el protocolo de configuración dinámica de host (DHCP; Dynamic Host Configuraron Protocol) [RFC 2131]. DHCP permite que un host obtenga (le sea asignada) una dirección IP de modo automático, así como cierta información adicional, como la dirección del router más cercano y la dirección de su servidor DNS.

Puesto que DHCP permite automatizar los aspectos relativos a conexión a la red, solemos referirnos a él como protocolo plug-and-play (conecta y funciona). Esta funcionalidad lo hace muy atractivo para los administradores de red, que de otra forma deberían realizar estas tareas manualmente. DHCP, al mismo tiempo, está ganando terreno en las redes de acceso residencial a Internet y en las LAN inalámbricas, donde los host se unen o se separan de la red frecuentemente.

El administrador de la red puede configurar DHCP de modo que cierto host reciba siempre una misma dirección IP, es decir, que cada vez que el host se una a la red reciba la misma dirección IP pre asignada. Pero muchas organizaciones e ISP residenciales no disponen de suficientes direcciones IP para todos sus host. Cuando esto ocurre, suele configurarse DHCP de modo que asigne a cada host que se conecta una dirección IP temporal. Por ejemplo, considere un ISP residencial con 2.000 clientes, aunque no pueda haber más de 400 de ellos conectados simultáneamente. Para tratar con sus 2.000 clientes, el ISP no necesita un bloque de 2.000 direcciones, sino que utilizando un servidor DHCP para asignar direcciones dinámicamente, sólo necesitará un bloque de 512 direcciones (por ejemplo, un bloque de la forma 200.23.30.0/23). Según los host se unen y se desligan, el servidor DHCP precisará actualizar su lista de direcciones IP disponibles. Cada vez que se une un host, el servidor DHCP asigna arbitrariamente una dirección de su depósito de direcciones disponibles; cada vez que se marcha un host, su dirección vuelve al depósito.

Otra razón importante por la que DHCP ha conseguido amplia difusión es la llegada de la computación móvil. Considere, por ejemplo, un estudiante que lleva un portátil de su dormitorio a la biblioteca o a un aula. Es muy probable que, en cada ocasión, el estudiante se conecte a una red distinta, y por tanto necesitará una nueva dirección IP en cada lugar. DHCP es especialmente adecuado en estas situaciones, en las que muchos usuarios vienen y se van, y sólo se requieren direcciones durante un corto periodo de tiempo.

***Rutado en internet***

Un conjunto de routers bajo el mismo control técnico y administrativo, y que ejecutan todos el mismo protocolo de rutado entre ellos, se conoce como sistema autónomo (SA). Cada SA, a su vez, contiene habitualmente varias redes. La diferencia más importante entre los protocolos de rutado de Internet está dada en si dichos protocolos se usan para rutar datagramas dentro de un único SA (intra-SA), o entre varios SA (inter-SA).

**Rutado Intra-sistema autónomo en Internet; RIP y OSF**

Los protocolos intra-SA se usan para determinar cómo rutar dentro de un sistema autónomo (SA; autonomous system). Los protocolos de rutado intra-SA también se conocen como protocolos de pasarela (gateway) interna o IGP. Históricamente, se han venido usando en mayor medida dos protocolos de rutado para el interior de los sistemas autónomos de Internet; RIP (Routing Information Protocol, protocolo de información de rutado) y OSPF (Open shortest Path First, primero el camino abierto más corto).

***RIP: protocolo de Información de rutado***

El protocolo de información de rutado (RIP) fue uno de los primeros protocolos de rutado intra-SA de Internet, y aún se sigue usando ampliamente. El amplio despliegue de RIP se debió en gran parte a su inclusión en 1982 en la distribución de UNIX de Berkeley (BSD; Berkeley Software Distribución) que ya soportaba TCP/IP. La versión 1 de RIP viene especificada en RFC 1058, existiendo una versión 2 compatible con la anterior que se documenta en RFC 2453.

RIP es un protocolo de vector de distancias. La versión de RIP especificada en RFC 1058 usa el recuento de saltos como una métrica del coste; es decir, cada enlace tiene un coste 1. El coste máximo de un camino está limitado a 15, por lo que se limita el alcance de RIP a sistemas autónomos que tengan siempre menos de 15 saltos de diámetro. Recuerde que en los protocolos de vector de distancias, los routers adyacentes intercambian información de rutado entre sí. En RIP, las actualizaciones de rutado se intercambian entre vecinos cada 30 segundos aproximadamente, empleando un mensaje de respuesta RIP. El mensaje de respuesta enviado por un router o host contiene una lista de hasta 25 redes destinos dentro del SA, así como la distancia del emisor a cada una de esas redes. Los mensajes de respuesta también se conocen como anuncios RIP.

A continuación veremos algunas cuestiones sobre los aspectos de la implementación de RIP. Recuerde que los routers RIP intercambian anuncios cada 30 segundos aproximadamente. Si un router no tiene noticia de su vecino en los siguientes 180 segundos, considerará que el vecino ya no está al alcance, es decir, que el vecino ha muerto o que el enlace de conexión se ha caído. Cuando esto ocurre, RIP modifica la tabla de encaminamiento local y propaga esta información, enviando anuncios a sus routers vecinos (aquéllos que todavía sean alcanzables).

También es posible que un router pida información a su vecino acerca del coste a cierto destino mediante un mensaje de petición RIP. Los routers se envían mensajes de petición y de respuesta uno a otro sobre UDP usando el puerto número 520. El paquete UDP va de un router a otro en un paquete IP estándar. El hecho de que RIP utilice un protocolo de capa de transporte (UDP) sobre un protocolo de capa de red (IP) para implementar una función de la capa de red (un algoritmo de rutado) puede parecer retorcido (¡y lo es!). Mirar un poco más de cerca cómo está implementado RIP nos aclarará las cosas.

Así, RIP es un protocolo de capa de aplicación que se ejecuta sobre UDP.

**OSPF: primero el camino abierto más corto.**

Como RIP, el rutado del camino abierto más corto (OSPF; Open Shortest Path First) sirve para el rutado intra-AS. El término "abierto" del nombre indica que la especificación del protocolo de rutado está públicamente disponible (al contrario, por ejemplo, que en el caso del protocolo EIGRP de Cisco). La versión más reciente de OSPF, la 2, está definida en el RFC 2328 (un documento público).

OSPF fue pensado para suceder a RIP, y por ello posee cierto número de características avanzadas. En el fondo, sin embargo. OSPF es un protocolo de estado de enlaces que utiliza un flujo de información de estado de enlaces y un algoritmo de Dijkstra de camino de coste mínimo. Con OSPF, cada router construye un mapa topológico (es decir, un grafo dirigido) del sistema autónomo completo. El router ejecuta localmente el algoritmo de Dijkstra del camino de coste mínimo para determinar un árbol de caminos de costes mínimos a todas las redes que parten de él como nodo raíz. La tabla de encaminamiento del router se obtiene a partir de este árbol de caminos de coste mínimo. Los costes de los enlaces vienen dados por el administrador de la red. El administrador podría elegir poner todos los enlaces con coste 1, con lo que se minimiza el número de saltos, o podría elegir ponerlos al inverso de su capacidad de canal, para minimizar el tráfico en los enlaces con menor ancho de banda. OSPF no establece políticas para fijar los pesos de los enlaces (esto es tarea del administrador de la red), sino que proporciona los mecanismos (protocolo) para determinar el rutado de coste mínimo para cierto conjunto de pesos de enlaces.

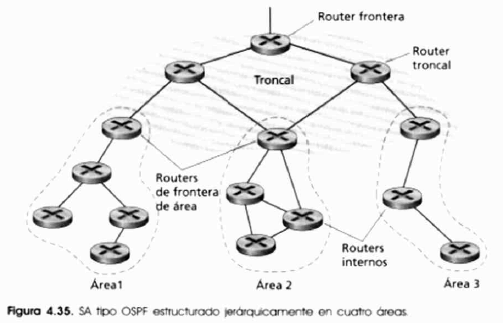
Con OSPF, un router difunde información de rutado al resto de los routers del sistema autónomo, y no sólo a sus routers vecinos. Cada router difunde información del estado de los enlaces cuando quiere que haya un cambio en el estado de un enlace (por ejemplo, un cambio en el coste, o un cambio en el estado activo inactivo). También difunde periódicamente el estado del enlace (al menos cada 30 minutos), aunque este no naya cambiado. RFC 2328 indica que “esta actualización periódica de los anuncios del estado de los enlaces añade robustez al algoritmo de estado de enlaces". Los anuncios OSPF se efectúan mediante mensajes OSPF que van directamente por IP, con un protocolo de capa superior de 89 para OSPF. Así, el protocolo OSPF debe implementar por sí mismo funcionalidades como la transferencia fiable de mensajes y la difusión del estado de enlaces. EI protocolo OSPF también comprueba que los enlaces sean operacionales (mediante un mensaje HELLO que se envía a cada vecino conectado), y permite a cada router OSPF obtener la base de datos completa del estado de la red desde un router vecino.

Algunos de los avances incorporados en OSPF son los siguientes:

* Seguridad, todos los intercambios entre los routers OSPF (por ejemplo, las actualizaciones de estados de enlaces) van autenticadas. Esto implica que sólo los routers fiables pueden participar en el protocolo OSPF del dominio, lo que previene que intrusos maliciosos puedan inyectar información incorrecta en las tablas de rutado.
* Múltiples caminos con el mismo coste, cuando hay varios caminos con un mismo coste, OSPF permite usar la multiplicidad de caminos (es decir, no hay por qué elegir un solo camino para todo el tráfico cuando el coste es el mismo).
* Soporte integrado para rutado por unidifusión o por multidifusión, "multicast OSPF" [RFC 1584J (OSPF con multidifusión) dispone de extensiones sencillas a OSPF para proporcionar rutado de multidifusión. MOSPF emplea la base de datos de enlaces existente en OSPF, y añade un nuevo tipo de anuncio de estado de enlaces al mecanismo existente de difusión de estado de enlaces OSPF.
* Soporte de jerarquía en un único dominio de rutado, es quizá el avance más significativo en OSPF es su capacidad de estructurar jerárquicamente un sistema autónomo, lo que le brinda a OSPF las ventajas de las estructuras de rutado jerárquico.

Cada sistema autónomo puede configurarse en áreas. Cada área ejecuta su propio algoritmo de rutado de estado de enlaces OSPF, donde cada router de un área difunde su estado de enlaces al resto de los routers de esa área. Los detalles internos del área son así invisibles para los routers de fuera del área. El rutado intraárea involucra sólo a aquellos routers de la misma área.

Dentro de cada área, uno o más routers de frontera de área son los responsables de rutar los paquetes hacia fuera. En el SA se configura exactamente un área como área troncal. La función primordial del área troncal es rutar el tráfico entre el resto de las áreas del SA. La troncal (backbone) siempre contiene todos los routers que lindan con las áreas del SA, y puede contener también otros routers no colindantes. El rutado interárea dentro del SA precisa que cada paquete sea rutado en primer lugar a un router de la frontera con la troncal (rutado intraárea), después a través de la troncal, hacia el router frontera del área destino, y de ahí a su destino final.



En la figura se muestra un diagrama de una red OSPF estructurada jerárquicamente. En ella podemos identificar cuatro tipos de routers OSPF:

* Routers internos, routers de fuera de áreas troncales que sólo pueden realizar rutado intra-SA.
* Routers de frontera de área, routers que pertenecen tanto a la frontera como a la troncal.
* Routers troncales (excepto frontera), routers que efectúan rutado en la troncal, aunque no son routers de frontera de área. Dentro de un área no troncal, los routers internos aprenden de la existencia de routers hacia otras áreas a través de la información (esencialmente un anuncio de estado de enlaces, aunque más bien sobre el coste de la ruta hacia otro área que sobre el coste del enlace en sí) difundida dentro del área por sus routers troncales.
* Routers de frontera, este intercambia información de rutado con los routers de otros sistemas autónomos. Este router podría, por ejemplo, emplear BGP para efectuar rutado inter-SA. Es mediante este tipo de routers de frontera como los otros routers aprenden sobre las rutas hacia redes externas.

OSPF es un protocolo relativamente complejo, y nuestro tratamiento aquí ha sido necesariamente breve.

**Rutado entre sistemas autónomos (inter-SA): BGP**

La versión 4 del protocolo de pasarela de frontera (BGP; Bordar Gateway Protocol), especificada en RFC 1771 (véase también [RFC 1772; RFC 1773]), es el actual estándar de facto para el protocolo de rutado entre dominios en uso en Internet. Suele denominarse BGP4, o simplemente BGP. Es un protocolo para el rutado entre sistemas autónomos (SA).



*La figura muestra los protocolos intra-SA e inter-SA usados dentro del SA2: BGP sirve para determinar las rutas a los destinos de fuera de SA2, mientras que OSPF permite determinar las rutas a los destinos de dentro del SA2.*

Aunque BGP tiene el mismo aspecto general que el protocolo de vector de distancias, es más apropiado caracterizarlo como un protocolo de vector de rutas. La razón es que los routers BGP vecinos, conocidos como pares (o iguales) BGP, intercambian información detallada sobre rutas (como la lista de SA en el camino hacia cierto destino) más que información de coste. Como en el protocolo genérico de vector de distancia, BGP es un protocolo distribuido en el que los routers BGP se comunican sólo con aquellos routers BGP directamente conectados. La información global sobre rutas hacia destinos distantes se propaga de forma SA a SA mediante el intercambio de información de rutado entre pares BGP directamente conectados. Es importante mencionar que BGP ruta a redes destino (en el sentido de direccionamiento), y no hacia máquinas destino o routers. Una vez que un datagrama llega a la red destino, el rutado del sistema autónomo de esa red se encargará de rutar el datagrama hacia su destino final. En BGP, cada sistema autónomo se identifica por su número de sistema autónomo (NSA; autonomous system number), que es globalmente único [RFC 1930]. Los números SA, como las direcciones IP, son asignados por los registros regionales del ICANN [ICANN 2002].

En el núcleo de BGP están los anuncios de ruta. Cada anuncio de ruta se envía entre pares BGP sobre conexiones punto a punto. Un anuncio consta de una dirección de red CIDRizada (por ejemplo, 128.119.40/24) y un conjunto de atributos relacionados con el camino para la red destino. Entre los atributos más importantes se encuentran dos: el atributo de camino (una lista explícita de todos los SA en el camino hacia la red destino especificada), y la identidad del router del próximo salto a lo largo del camino a la red destino.

El procedimiento BGP gira en torno a tres actividades, y las tres involucran anuncios de ruta:

* Recepción y filtrado de anuncios de ruta desde el (los) vecino(s) directamente conectado(s), cada router BGP recibirá anuncios de ruta desde un par BGP. Podemos ver cada anuncio de ruta BGP recibido como una promesa. Cada par BGP que anuncia una ruta a un SA destino promete que si un SA adyacente encamina un datagrama para ese SA destino, será capaz de encaminar dicho datagrama hacia ese destino. Cada router BGP puede también filtrar (desechar) los anuncios de ruta. Por ejemplo, un router BGP ignorará los anuncios que contengan su propio número SA en el atributo SA-PATH (camino SA), puesto que de seguir ese camino resultaría un bucle. Puesto que se especifica el camino SA completo, cada administrador de red puede ejercer un control importante sobre por dónde se rutan sus datagramas. Por ejemplo, resulta posible para un SA como Montescos.net implementar cierta política como que “el tráfico de mi SA no deberá cruzar el SA Capuletos.net”.
* Selección de ruta, un router BGP puede recibir diversos anuncios de ruta hacia el mismo SA destino, y debe elegir cuál de las rutas usará entre las anunciadas. El SA destino y el siguiente salto para el camino elegido deben estar instalados en las tablas de encaminamiento del router. Un router BGP podría conocer muchas rutas hacia un destino dado, pero usualmente sólo instalará un router para el próximo salto para ese destino en la tabla de encaminamiento. Pero, ¿cómo elige el router BGP entre los diversos caminos? BGP distingue claramente entre mecanismo de rutado y política de rutado. Concretamente, BGP no dice cómo debe elegir el SA entre los caminos anunciados. Ésta es una decisión de política que se deja al administrador de red del SA. Cada administrador de red podrá especificar lo que se denominan preferencias locales; por ejemplo, la indicación de que siempre es preferible rutar a través del SA A que a través del SA B, siempre que sea posible elegir. En ausencia de preferencias locales, la ruta seleccionada es siempre la ruta con el camino SA más corto (es decir, la que atraviesa el menor número de sistemas autónomos en su camino al destino).
* Envío de anuncios de ruta al vecindario, al igual que los routers BGP reciben anuncios de ruta de sus vecinos, también pueden anunciar sus rutas. De nuevo, BGP proporciona un mecanismo (aunque no una política) para anunciar. Esto da al administrador de red un importante grado de control sobre el tráfico que será rutado al interior de su red. Volviendo a nuestro ejemplo de Montescos/Capuletos, será fácil para los Mónteseos evitar que los Capuletos envíen tráfico directamente hacia la red de los Mónteseos. Supongamos que los Capuletos son los vecinos de al lado de los Mónteseos; bastará con que los Mónteseos no anuncien ruta alguna a los Capuletos que contenga la red de los Mónteseos. Pero el restringir el tráfico controlando únicamente los anuncios de ruta SA puede ser sólo parcialmente efectivo. Por ejemplo, si los Gómez están entre los Mónteseos y los Capuletos, y los Mónteseos anuncian rutas a los Gómez que pasan a través de los Montescos, los Mónteseos no podrán prevenir (con los mecanismos BGP) que los Gomez anuncien dichas rutas a los Capuletos.

Los pares BGP se comunican usando el protocolo TCP y el puerto 179. TCP proporciona un mecanismo de intercambio de mensajes fiable y de congestión controlada entre pares BGP. Recuerde que, por contra, como vimos anteriormente, los pares RIP se comunican mediante UDP no fiable, y que OSPF utiliza su propio protocolo para intercambiar mensajes OSPF. El protocolo BGP define los cuatro tipos de mensajes siguientes: OPEN, UPDATE, KEEPALIVE, y NOTIF1CATION.

En la práctica: ¿POR QUÉ HAY TRES PROTOCOLOS DE RUTADO INTER-SA E INTRA-SA DIFERENTES?

Habiendo ya estudiado los detalles de los protocolos específicos inter-SA e intra-SA desplegados en la Internet actual, abordemos en primer lugar la que es quizá la pregunta fundamental sobre estos protocolos (es de esperar que haya estado preguntándose esto todo este tiempo, ¡y no se haya perdido en los detalles!): ¿por qué se emplean tres protocolos de rutado Inter-SA e intra-SA diferentes?. La respuesta a esta pregunta se encuentra en el núcleo de las diferencias entre los objetivos del rutado dentro de un SA y entre varios SA:

* Política, entre varios SA, predominan cuestiones de política. Puede que lo importante sea que el tráfico que se origina en cierto SA no sea capaz de pasar a través de otro SA concreto. Igualmente, cierto SA pudiera querer controlar qué tráfico en tránsito lleva entre otros SA. Hemos visto que BGP lleva específicamente atributos de camino, y proporciona información de rutado de forma controlada y distribuida, de forma que sean posibles tales decisiones de política de rutado. Dentro de un SA todo se encuentra nominalmente bajo el mismo control administrativo, y así, las cuestiones políticas juegan un papel mucho menos importante en la elección de las rutas internas al SA.
* Escala, la posibilidad de escalado del algoritmo y sus estructuras de datos de rutado de forma que se pueda manejar el rutado hacia/entre un gran número de redes es una cuestión critica en el rutado inter-SA. Dentro de un SA, la escalabilidad es menos importante; salvo un detalle, si un único dominio administrativo se vuelve demasiado grande, siempre será posible dividirlo en dos SA y realizar un rutado inter-SA entre los dos SA nuevos (recuerde que OSPF permite construir dicha jerarquía mediante la división de un SA en áreas).
* Prestaciones, dado que el rutado inter-SA está tan políticamente orientado, la calidad (por ejemplo, las prestaciones) de las rutas usadas es una cuestión menor (es decir, una ruta más larga o más costosa que satisfaga cierta política podría predominar sobre una ruta que fuera más corta pero que no cumpliera los criterios). Ciertamente, hemos visto que entre SA ni siquiera hay noción de coste (diferente a la de recuento de SA) asociada a las rutas. Dentro de un único SA, sin embargo, estos criterios políticos son menos Importantes, lo que permite al rutado concentrarse en el nivel de prestaciones obtenido por una ruta.

***Mascara o Mask***, es una combinación de [bits](https://es.wikipedia.org/wiki/Bits) que sirve para delimitar el ámbito de una [red de ordenadores](https://es.wikipedia.org/wiki/Red_de_ordenadores) (local o externo). Permite delimitar la parte de la [dirección IP](https://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) es el número de la red, incluyendo la subred, y qué parte es la correspondiente al [host](https://es.wikipedia.org/wiki/Host), también permite identificar si el destino de un paquete está en la red a la que pertenece el emisor (red local) o si es una red externa. La máscara determina todos los parámetros de una subred: dirección de red, dirección de difusión (broadcast) y direcciones asignables a nodos de red ([hosts](https://es.wikipedia.org/wiki/Host)).

***SuperRed, SuperNet o Supernetting (agregación de rutas)***, es una red [IP](https://es.wikipedia.org/wiki/Internet_Protocol) que está formada por la combinación de dos o más redes o subredes con un prefijo [CIDR](https://es.wikipedia.org/wiki/CIDR) común. El prefijo de enrutado de la superred comprende los prefijos de las redes que la constituye. No debe contener prefijos de otras redes que no estén en el mismo camino de enrutado. El proceso de formar una supernet es denominado supernetting o agregación de rutas.

Supernetting en [Internet](https://es.wikipedia.org/wiki/Internet) sirve como estrategia preventiva para evitar fragmentación topológica del espacio de [direccionamiento IP](https://es.wikipedia.org/wiki/Direcci%C3%B3n_IP), utilizando un sistema de asignación jerárquico, que delega el control de segmentos del espacio de direcciones a los proveedores regionales del servicio de red. Este método facilita la agregación de las rutas por regiones.

***Subred, SubNet o Subnetting (desagregación de ruta)***, es una técnica mediante la cual se puede dividir una red en subredes de tamaño fijo (todas las subredes tienen el mismo tamaño), con el fin de incrementar el espectro de redes disponibles por el cliente, combatir la carencia de direcciones IP y suplir necesidades de las organizaciones de administrar su propia estructura de red. Sin embargo, por la escasez de [direcciones IP](https://es.wikipedia.org/wiki/Direcci%C3%B3n_IP), hoy en día frecuentemente se usan [subredes de tamaño variable](https://es.wikipedia.org/wiki/VLSM).

***VLSM (Virtual Length Subnet Masking),*** representan otra de las tantas soluciones que se implementaron para evitar el agotamiento de direcciones IP (1987), como la división en subredes, el enrutamiento sin clases [CIDR](https://es.wikipedia.org/wiki/CIDR), [NAT](https://es.wikipedia.org/wiki/NAT) y las direcciones IP privadas.

Su función es descentralizar las redes y de esta forma conseguir redes más seguras y jerárquicas y a su vez evitar el desperdicio de direcciones. Un ejemplo: Si se utiliza una [máscara de subred](https://es.wikipedia.org/wiki/M%C3%A1scara_de_subred) de tamaño fijo (la misma máscara de subred en todas las subredes), todas las [subredes](https://es.wikipedia.org/wiki/Subred) van a tener el mismo tamaño. Por ejemplo, si la subred más grande necesita 200 hosts, todas las subredes van a tener el mismo tamaño de 256 direcciones IP (nota: se ha redondeado hacia arriba, hacia la siguiente potencia, de 2). Si una subred que necesita 10 equipos, se asigna la misma subred de 256 direcciones, aunque las restantes 246 direcciones no se utilicen.

El concepto básico de VLSM es muy simple: se toma una red y se divide en subredes fijas, luego se toma una de esas [subredes](https://es.wikipedia.org/wiki/Subred) y se vuelve a dividir, tomando bits "prestados" de la porción de hosts, ajustándose a la cantidad de hosts requeridos por cada segmento de nuestra red.

Por ejemplo, si se toma la dirección de red 192.168.1.0/24 y se subdivide usando una máscara /26 tendremos 4 subredes (192.168.1.0/26, 192.168.1.64/26, 192.168.1.128/26 y 192.168.1.192/26). Suponga que se construye un enlace serie entre dos routers y tomamos para ello una de las subredes (la 192.168.1.0/26): con esta máscara de subred sin aplicar vlsm se desperdiciarían 60 direcciones utilizables (26-2=62 menos las 2 direcciones aplicadas a las interfaces de los routers da 62 hosts, [64-2=62] una dirección para el nombre de la red o dirección de red y la otra para la dirección de difusión o broadcast).

Ahora, si se aplica vlsm a la subred anterior (la 192.168.1.0/26) y se toman "prestados" 4 bits de la porción de host tendríamos otras 16 subredes /30 (192.168.1.0/30, 192.168.1.4/30, 192.168.1.8/30, 192.168.1.12/30, 192.168.1.16/30 y así sucesivamente hasta la 192.168.1.60/30) cada una con un total de 4 direcciones totales pero solamente dos direcciones utilizables y no se genera desperdicio. Finalmente podemos tomar cualquiera de ellas, por ejemplo la 192.168.1.4/30 y aplicar las direcciones 192.168.1.5/30 y 192.168.1.6/30 a las interfaces de los routers.

*Unidad 5 - Capa de Enlace*

***Función y servicios***

La capa de enlace es la responsable de transferir datagramas hacia el otro lado de un enlace determinado. El protocolo de la capa de enlace define el formato de los paquetes que se intercambian entre los nodos en los extremos de ese enlace, como también las acciones a tomar por esos nodos cuando envían y reciben paquetes.

Un protocolo de la capa de enlace tiene el trabajo de mover un datagrama de la capa de red sobre un solo enlace del camino. Una característica importante de la capa de enlace es que un datagrama puede ser manejado por protocolos diferentes de la capa de enlace sobre los diferentes enlaces en el camino. Por ejemplo, un datagrama puede ser manejado por Ethernet en el primer enlace, PPP en el último enlace y frame relay en todos los enlaces intermedios.

El servicio básico de la capa de enlace es mover un datagrama desde un nodo a otro adyacente sobre un solo enlace de comunicación. Pero los detalles del servicio de la capa de enlace dependen del protocolo específico que es empleado sobre ese enlace. Los posibles servicios que puede ofrecer un protocolo de la capa de enlace son:

• *Acceso al medio y framing*: casi todos los protocolos de la capa de aplicación encapsulan cada datagrama de la capa de red dentro de un marco (frame) de la capa de enlace antes de transmitirlo por el enlace. Un frame consiste de un campo de datos, en el cual el datagrama de la capa de red es insertado, y un número de campos de header. Un protocolo de enlace de datos especifica la estructura del frame, como también un protocolo de acceso al medio que especifica las reglas de transmisión.

• *Transmisión confiable:* Si un protocolo de la capa de enlace (como TCP) provee el servicio de transmisión confiable, este garantiza que los datagramas de la capa de red atravesarán el enlace sin errores. Recordemos que los protocolos de la capa de transporte también pueden proveer un servicio de transmisión confiable. En forma similar a un servicio de transporte confiable de la capa de transporte, un servicio de transmisión confiable de la capa de enlace se logra con reconocimientos y retransmisiones.

• *Control de flujo:* Los nodos en cada lado de un enlace tienen una cantidad límite de paquetes a almacenar en buffer. Este es un problema, cuando un nodo receptor puede recibir frames más rápido de lo que lo pude procesar. Sin control de flujo, el buffer del receptor puede sobrecargarse y pueden perderse frames. De forma similar a la capa de transporte, un protocolo de capa de enlace puede proveer control de flujo para prevenir que el transmisor bombardee al receptor.

• *Detección de errores:* Un nodo receptor puede interpretar incorrectamente un bit en un frame como un cero cuando se ha transmitido un uno (y viceversa). Estos errores son introducidos por la atenuación de la señal y ruido electromagnético. Debido a que no es necesario pasar un datagrama con errores, varios protocolos de la capa de enlace proveen un mecanismo para que un nodo detecte la presencia de uno o más errores. Para esto en nodo transmisor debe setear bits de detección de errores en el frame, debiendo el receptor realizar un chequeo de errores. La detección de errores es un servicio muy común en protocolos de la capa de enlace. La capa de transporte y de red también proveen protocolos limitados de detección de errores. En la capa de enlace estos protocolos son más sofisticados y se implementan en hardware.

• *Corrección de errores:* La corrección de errores es similar a la detección de errores, excepto que un receptor puede no solo detectar los errores introducidos en el frame sino también determinar exactamente donde ha ocurrido el error (y en consecuencia corregir dicho error).

• *Half-Duplex y Full-Duplex:* Con transmisión full-duplex, ambos nodos en los extremos del enlace pueden transmitir paquetes al mismo tiempo. Con transmisión half-duplex, un nodo no puede transmitir y recibir al mismo tiempo.

Como se ha indicado anteriormente, muchos de los servicios proporcionados por la capa de enlace tienen fuerte paralelismo con los servicios proporcionados en la capa de transporte. Por ejemplo, tanto la capa de enlace como la de transporte pueden proporcionar entrega fiable. Aunque los mecanismos utilizados para proporcionar entrega fiable en las dos capas son similares, los dos servicios de entrega fiable no son iguales. Un protocolo de transporte proporciona entrega fiable entre dos procesos sobre una base terminal-a-terminal; un protocolo fiable de la capa de enlace proporciona el servicio de entrega fiable entre dos nodos conectadas por un único enlace (direccionamiento físico). De forma semejante, ambos protocolos de la capa de enlace y de transporte pueden proporcionar control de flujo y detección de errores; de nuevo, el control de flujo en un protocolo de la capa de transporte se proporciona en una base terminal-a-terminal, mientras que en un protocolo de la capa de enlace se proporciona en una base de nodo a nodo adyacente.

***Identificación entre dos máquinas en una red Ethernet***

Las máquinas en una red Ethernet se identifican con su número de MAC que es único por adaptador. Las direcciones Ethernet o MAC son de 48 bits de longitud y se encuentran en Hardware. De estos 48 bits se utilizan los 3 bytes más significativos para identificar al fabricante y los 3 bytes menos significativos para identificar la placa. La principal característica es que es única. Una diferencia entre una dirección Mac y una IP jerárquica es que la MAC es portable, ya que puedo mover la tarjeta LAN a otra, y la IP no es portable, ya que depende de la subred IP a la que el nodo está conectado.

La dirección de Broadcast de la capa de enlace es la FF:FF:FF:FF:FF:FF, esto se utiliza cuando la maquina A quiere enviar datagrama a otra host B que esta fuera su red, y A en su tabla ARP no tiene la MAC de B, entonces A realiza un Broadcast de un paquete ARP query, conteniendo la IP de B, con Dirección MAC destino: FF-FF-FF-FF-FF-FF. Todas las maquinas en la LAN reciben el ARP query. B Recibe el paquete ARP y responde a A con su dirección MAC (ARP Reply). La trama es enviada a la dirección MAC de A (unicast). A guarda el par de direcciones IP-MAC en su tabla ARP hasta que la información se considere vieja (timeout).

ARP es “plug-and-play”: Los nodos crean (bajo demanda) sus tablas ARP sin intervención del administrador de la red

Funcionamiento I

Si A quiere enviar una trama a la [dirección IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) de B (misma red), mirará su tabla ARP para poner en la trama la dirección destino física correspondiente a la IP de B. De esta forma, cuando les llegue a todos la trama, no tendrán que deshacerla para comprobar si el mensaje es para ellos, sino que se hace con la dirección física.

Funcionamiento II

Si A quiere enviar un mensaje a C (un nodo que no esté en la misma red), el mensaje deberá salir de la red. Así, A envía la trama a la dirección física de salida del router. Esta dirección física la obtendrá a partir de la [IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) del router, utilizando la tabla ARP. Si esta entrada no está en la tabla, mandará un mensaje ARP a esa [IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) (llegará a todos), para que le conteste indicándole su dirección física.

Una vez en el router, éste consultará su tabla de encaminamiento, obteniendo el próximo nodo (salto) para llegar al destino, y saca el mensaje por la interfaz correspondiente. Esto se repite por todos los nodos, hasta llegar al último router, que es el que comparte el medio con el host destino. Aquí el proceso cambia: la interfaz del router tendrá que averiguar la dirección física de la [IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) destino que le ha llegado. Lo hace mirando su tabla ARP, y en caso de no existir la entrada correspondiente a la [IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP), mandará un mensaje ARP a esa [IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) (llegará a todos), para que le conteste indicándole su dirección física

***Algoritmo de acceso al medio en Ethernet.***

Una red [Ethernet](http://es.wikipedia.org/wiki/Ethernet) se comporta como un medio compartido, es decir, sólo un dispositivo puede transmitir con éxito por cada vez, y cada uno es responsable de la detección de colisiones y de la retransmisión, con acceso aleatorio: si dos nodos transmiten a la vez hay una colisión de la que deben recuperarse. Deben coordinarse utilizando el mismo canal que para transmitir (es in-line).

***CSMA/CD:*** La técnica de control de acceso al medio más usada para topologías bus/árbol y estrella es la técnica de Acceso Múltiple sensible a la portadora con detección de colisiones (CSMA/CD - Carrier sense Multiple Access with Collision Detect) y tiene las siguientes características:

• Tiene como antecedentes a ALOHA, ALOHA ranurada y CSMA.

• Permite el acceso múltiple mediante “contienda” donde todas las estaciones compiten, no hay control para la asignación de turnos.

• Las colisiones se producen cuando dos o más estaciones transmiten simultáneamente. Esta situación es detectada y corregida más tarde.

***Algoritmo CSMA/CD:***

• Se comprueba el estado del canal antes de iniciar una transmisión, si el canal esta libre se transmite de inmediato (CSMA/CD -carrier sense)

• Si el canal está ocupado espera a que este libre

• Mientras se transmite comprueba si se produce una colisión (CSMA/CD -colisión detección)

• Si se produce una colisión se interrumpe la transmisión.

***Algoritmo de control de Colisiones:***

• Después de la primer colisión se espera 0 o 1 intervalo de tiempo.

• Después de la segunda colisión se espera 0, 1, 2, o 3 intervalos M aleatoriamente.

• Después de la colisión i se elige un número entre o y 2i -1

• El número máximo de intentos es 16. Después de esto se reporta al host.

El protocolo de Ethernet exige que la detección de colisión se haga durante la transmisión de la trama.

Ethernet provee servicios no orientados a la conexión a la capa de red. Es decir, cuando el adaptador A desea enviar un datagrama al adaptador B, el adaptador A encapsula el datagrama en un frame Ethernet y envía el frame por la LAN, sin establecer la conexión con el adaptador B. Y NO es confiable. Eficiente con baja carga -habrá pocas colisiones-.

***CSMA/CD: protocolo de acceso múltiple de Ethernet***

Los nodos de una LAN Ethernet están interconectados por un canal de difusión, por lo que cuando un adaptador transmite un marco, todos los adaptadores de la LAN reciben el marco. Ethernet utiliza un algoritmo de acceso múltiple CSMA/CD.

Estos mecanismos dan a CSMA/CD unas prestaciones mucho mejores que las de ALOHA con franjas en un entorno LAN. Esto requiere que cada adaptador Ethernet sea capaz de (1) darse cuenta de cuándo está transmitiendo otro adaptador y (2) detectar una colisión mientras está transmitiendo. Los adaptadores Ethernet realizan estas dos tareas midiendo los niveles de voltaje antes y durante la transmisión.

***¿Qué es un bridge (o puente)? ¿Y un switch? ¿En qué se diferencian?***

Un switch/bridge es un dispositivo de capa de enlace, que permite interconectar segmentos de LAN formando dominios de colisión separados, pero un mismo dominio de broadcast.  
Al recibir una trama por una interfaz, consulta una tabla interna que construye automáticamente con MAC/interfaz, y reenvía la trama por la interfaz adecuada (si es la misma interfaz de entrada, no hace nada; si no estaba en la tabla, hace flooding reenviando la trama a todas las interfaces excepto por la que entró).  
 El bridge conmuta por software y está pensado para conmutar entre redes, por lo que tienen pocas interfaces (2). Normalmente manejan de a una trama por vez. Utilizan store-and-fordware (mantienen buffers internos en los que almacenan las tramas de entrada, retransmitiéndolas recién cuando ingresaron por completo).  
 El switch conmuta por hardware -más rápido- y se pensó para conectarse a hosts directamente. Pueden configurarse en modo store-and-fordware o cut-through (comienza a enviar el frame ni bien reconoce la MAC destino). Pueden manejan múltiples tramas simultáneamente y mantener caminos paralelos. Los switchs tienen generalmente cada interfaz directamente conectada a un host, con lo que se logra una transferencia full-duplex Tanto bridges como switchs no tienen direcciones LAN (MAC). Es decir, son “transparentes”.  
  
 ***HUB***

**Concentradores (Hubs):** El término hub describe la manera en que las conexiones de cableado de cada nodo de una red se centralizan y conectan en un único dispositivo. Se suele aplicar a concentradores Ethernet, Token Ring, y FDDI (Fiber Distributed Data Interface) soportando módulos individuales que concentran múltiples tipos de funciones en un solo dispositivo. Normalmente los hubs incluyen ranuras para aceptar varios módulos y un panel trasero común para funciones de encaminamiento, filtrado y conexión a diferentes medios de transmisión (por ejemplo Ethernet y TokenRing).

A un hub Ethernet se le denomina "repetidor multipuerta". El dispositivo repite simultáneamente la señal a múltiples cables conectados en cada uno de los puertos del hub. En el otro extremo de cada cable está un nodo de la red, por ejemplo un ordenador personal. Un hub Ethernet se convierte en un hub inteligente (smart hub) cuando puede soportar inteligencia añadida para realizar monitorización y funciones de control.

Los concentradores inteligentes (smart hub) permiten a los usuarios dividir la red en segmentos de fácil detección de errores a la vez que proporcionan una estructura de crecimiento ordenado de la red. La capacidad de gestión remota de los hubs inteligentes hace posible el diagnóstico remoto de un problema y aísla un punto con problemas del resto de la RAL, con lo que otros usuarios no se ven afectados.

A un hub TokenRing se le denomina Unidad de Acceso Multiestación (MAU - Multiestation Access Unit). Las MAUs se diferencian de los hubs Ethernet porque las primeras repiten la señal de datos únicamente a la siguiente estación en el anillo y no a todos los nodos conectados a ella como hace un hub Ethernet. Las MAUs pasivas no tienen inteligencia, son simplemente retransmisores. Las MAUs activas no sólo repiten la señal, además la amplifican y regeneran. Las MAUs inteligentes detectan errores y activan procedimientos para recuperarse de ellos.

***¿Qué es una Colisión?***

Un dominio de colisión es un segmento físico de una [red de computadores](http://es.wikipedia.org/wiki/Red_de_computadores) donde es posible que las tramas puedan "colisionar" (interferir) con otros. Estas colisiones se dan particularmente en el protocolo de red [Ethernet](http://es.wikipedia.org/wiki/Ethernet).

A medida que aumenta el número de nodos que pueden transmitir en un segmento de red, aumentan las posibilidades de que dos de ellos transmitan a la vez. Esta transmisión simultánea ocasiona una interferencia entre las señales de ambos nodos, que se conoce como colisión. Conforme aumenta el número de colisiones disminuye el rendimiento de la red.

***¿Qué diferencia un HUB de un Switch?***

*Switches*: Los switches tienen la funcionalidad de los hubs a los que añaden la capacidad principal de dedicar todo el ancho de banda de forma exclusiva a cualquier comunicación entre sus puertos. Esto se consigue debido a que el conmutador no actúa como repetidor multipuerto, sino que únicamente envía paquetes de datos hacia aquella puerta a la que van dirigidos. Esto es posible debido a que los equipos configuran unas tablas de encaminamiento con las direcciones MAC (nivel 2 de OSI) asociadas a cada una de sus puertas.

De esta forma se consigue que el tráfico interno en los distintos segmentos de red conectados al conmutador no afecte al resto de la red aumentando de esta manera la eficiencia de uso del ancho de banda. Hay tres tipos de conmutadores o técnicas de conmutación:

• Almacenar - Transmitir.

• Cortar - Continuar.

• Híbridos.

La diferencia entre el hub y el switch es que los switches toman decisiones basándose en las direcciones MAC y los hubs no toman ninguna decisión. Como los switches son capaces de tomar decisiones, así hacen que la LAN sea mucho más eficiente. Los switches hacen esto "conmutando" datos sólo desde el puerto al cual está conectado el host correspondiente. A diferencia de esto, el hub envía datos a través de todos los puertos de modo que todos los hosts deban ver y procesar (aceptar o rechazar) todos los datos. Esto hace que la LAN sea más lenta.

***Dispositivos dividen dominios de broadcast? ¿Y dominios de colisión?***

Un dominio de difusión (Broadcast) es un área lógica en una [red](http://es.wikipedia.org/wiki/Red_de_ordenadores) en la que cualquier ordenador conectado a la red puede transmitir directamente a cualquier otro en el dominio sin precisar ningún dispositivo de encaminamiento, dado que comparten la misma subred, dirección de puerta de enlace y están en la misma [VLAN](http://es.wikipedia.org/wiki/VLAN) (VLAN por defecto o instalada).

Un dominio de colisión es un segmento físico de una [red de computadores](http://es.wikipedia.org/wiki/Red_de_computadores) donde es posible que las tramas puedan "colisionar" (interferir) con otros. Estas colisiones se dan particularmente en el protocolo de red [Ethernet](http://es.wikipedia.org/wiki/Ethernet).

A medida que aumenta el número de nodos que pueden transmitir en un segmento de red, aumentan las posibilidades de que dos de ellos transmitan a la vez. Esta transmisión simultánea ocasiona una interferencia entre las señales de ambos nodos, que se conoce como colisión. Conforme aumenta el número de colisiones disminuye el rendimiento de la red.

Un dominio de colisión puede estar constituido por un solo segmento de cable Ethernet en una Ethernet de medio compartido, o todos los nodos que afluyen a un concentrador Ethernet en una Ethernet de par trenzado, o incluso todos los nodos que afluyen a una red de concentradores y repetidores.

Para segmentar dominios de colisión, se utilizan Routers, y Switch/Bridge.

El Hub no genera ningún tipo de división (ni de colisión, ni de broadcast), ya que es una especie de “extensor” de la señal que pertenece a la capa física.

***Finalidad del protocolo ARP***

Protocolo de capa 2, Mapea direcciones MAC a direcciones IP, es un Helpper. Funciones Request y replay. Dentro de una misma red, las máquinas se comunican enviándose tramas físicas. Las tramas Ethernet contienen campos para las direcciones físicas de origen y destino (6 bytes cada una):

El problema que se nos plantea es cómo podemos conocer la dirección física de la máquina destino. El único dato que se indica en los datagramas es la dirección IP de destino. ¿Cómo se pueden entregar entonces estos datagramas? Necesitamos obtener la dirección física de un ordenador a partir de su dirección IP. Esta es justamente la misión del protocolo ARP (Address Resolution Protocol, protocolo de resolución de direcciones).

ARP está documentado en el Request for comments RFC 826.

El protocolo [RARP](http://es.wikipedia.org/wiki/RARP) realiza la operación inversa y se encuentra descrito en el [RFC 903](http://tools.ietf.org/html/rfc903).

En [Ethernet](http://es.wikipedia.org/wiki/Ethernet), la capa de enlace trabaja con direcciones físicas. El protocolo ARP se encarga de traducir las direcciones [IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) a direcciones [MAC](http://es.wikipedia.org/wiki/MAC) (direcciones físicas). Para realizar esta conversión, el nivel de enlace utiliza las tablas ARP (cuyas direcciones tienen un tiempo de vida “TTL”), cada interfaz tiene tanto una [dirección IP](http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) como una dirección física [MAC](http://es.wikipedia.org/wiki/MAC).

ARP se utiliza en 4 casos referentes a la comunicación entre 2 hosts:

1. Cuando 2 hosts están en la misma red y uno quiere enviar un paquete a otro.
2. Cuando 2 host están sobre redes diferentes y deben usar un gateway/router para alcanzar otro host.
3. Cuando un router necesita enviar un paquete a un host a través de otro router.
4. Cuando un router necesita enviar un paquete a un host de la misma red.

Si la MAC no se encuentra en la tabla, hay 2 opciones:

* Que el destino sea del mismo segmento/subred (1 y 4): Se averigua la MAC con una consulta ARP al broadcast físico indicando la IP, y el adaptador que reconoce la IP destino responde con su MAC, que es agregada a la tabla del emisor.
* Que esté en un segmento de red diferente (2 y 3) (al aplicar la máscara de (sub)red, se obtiene una dirección distinta a la del origen): En este caso se envía el frame al gateway, que luego continúa retransmitiéndolo. El datagrama en el interior del frame contiene la IP del destino, pero el frame contiene la MAC del gateway (si no la conoce, utiliza ARP para solicitarla). Al enviarse el frame, es recibido por el router, sube a su capa de red y al ver una IP destino distinta de la suya, vuelve a bajar a enlace para ser retransmitida al próximo router que corresponda según la tabla de ruteo. Eventualmente un router conocerá al equipo destino, y le enviará el datagrama con la MAC correcta -si no la conoce, utiliza ARP-.

Para todo esto se necesitan tablas ARP, y la filosofía que seguiríamos es la misma que tendríamos para localizar al señor "X" entre 150 personas: preguntar por su nombre a todo el mundo, y el señor "X" nos responderá. Así, cuando a "A" le llegue un mensaje con dirección origen IP y no tenga esa dirección en su caché de la tabla ARP, enviará su trama ARP a la dirección broadcast (física = FF:FF:FF:FF:FF:FF), con la IP de la que quiere conocer su dirección física. Entonces, el equipo cuya dirección IP coincida con la preguntada, responderá a "A" enviándole su dirección física. En este momento "A" ya puede agregar la entrada de esa IP a la caché de su tabla ARP. Las entradas de la tabla se borran cada cierto tiempo, ya que las direcciones físicas de la red pueden cambiar (Ej: si se estropea una tarjeta de red y hay que sustituirla, o simplemente algún usuario de la red cambia de dirección IP).

**Trama Ethernet:** (mac origen: MAC ORIGEN mac destino: FF-FF-FF-FF-FF-FF)

**Solicitud ARP:** (mac origen: MAC ORIGEN ip origen: IP ORIGEN

mac destino: 00-00-00-00-00 ip destino: <IP default GW>)

*Comprobación de paridad:* Es el esquema más sencillo para detectar errores y consiste en añadir un bit de paridad al final del bloque de datos. El valor de este bit se determina de tal forma que el carácter resultante tenga un número impar de unos (paridad impar) o un numero par (paridad par). Si se utiliza paridad impar y se invierte uno o un número impar de bits erróneamente durante la transmisión, el receptor detectara un error. Sin embargo si se invierten dos o un numero par de bits aparecerá un error no detectado. Algo similar ocurre si se usa paridad par.

Típicamente la paridad par se usa para la transmisión sincrónica y la impar para la asincrónica.

*Comprobación de Redundancia Cíclica (CRC “Cyclic Redundancy Check”)*: Es uno de los mecanismos más comunes y potente. Dado un bloque o mensaje de k bits el transmisor genera una secuencia de n bits denominada secuencia de comprobación de la trama (FCS) de tal manera que la trama resultante con n+k bits sea divisible por algún número determinado.

El receptor dividirá la trama recibida por ese número y si no hay resto en la división se supone que no ha habido errores. Se puede resolver el procedimiento de tres maneras: usando aritmética de módulo 2, usando polinomios o usando lógica digital. Si usamos polinomios podemos decir que FCS es el resto de la división de: (Mensaje \* Xr) / Polinomio Generador, donde r es el grado del polinomio generador.

***IEEE 802.3. Existen diferencias con Ethernet?***

IEEE 802.3 es una colección de estándares que definen la capa física y de enlace de

Ethernet. Específicamente, el estándar 802.3 puro, define la tecnología 10Base5, en la cual se utiliza cable coaxil especial. Posee una velocidad de transferencia 10 mbit/s, y permite una longitud de cable máxima de 500 metros entre nodos.

***Comparación Ethernet/IEEE 802.3***

Ambas LANs, Ethernet y IEEE 802.3 son redes de difusión, lo que significa que todas las estaciones ven todos los paquetes, sin tener en cuenta si representan un destino determinado. Cada estación debe examinar los paquetes recibidos para determinar si la estación es un destino. En este caso, el paquete se pasa a una capa de protocolo superior para su procesamiento adecuado.

Las diferencias entre LANs Ethernet y IEEE 802.3 son sutiles. Ethernet proporciona servicios correspondientes a las capas 1 y 2 del modelo de referencia OSI, mientras que IEEE 802.3 especifica la capa física (Capa 1) y la parte de acceso-canal de la capa de enlace (Capa 2), pero no define un protocolo de control de enlace lógico. Así como el resto de funciones de las capas 1 y 2, tanto Ethernet como IEEE 802.3 están implementadas en hardware, en general a través de una tarjeta de interface en un ordenador o a través de una placa principal en el propio ordenador.

Ethernet es muy similar a IEEE 802.3 10BaseT. Ambos protocolos especifican una red de topología de bus con un cable de conexión entre las estaciones finales y el soporte de red actual. En el caso Ethernet, es cable se denomina cable transceptor. El cable transceptor conecta a un dispositivo transceptor conectado al soporte físico de la red.

La configuración IEEE 802.3 es prácticamente la misma, a excepción de que el cable de conexión se denomina unidad de conexión del soporte (MAU). En ambos casos, el cable de conexión se conecta a la placa del interface (o a un circuito de interface) dentro de la estación final.

***No está en la práctica. Describa el algoritmo de acceso al medio empleado por 802.5***

El protocolo MAC mas usual en redes LAN con topología en anillo es el de paso de testigo utilizado por la normalización 802.5.

Control de acceso al medio en IEEE 802.5 (Token Ring): Este método consiste en que existe una trama pequeña llamada testigo, que circula por la red cuando no hay ninguna estación transmitiendo. Cuando una estación desea transmitir, cuando le llega el testigo, lo agarra, le cambia un cierto bit y le añade la trama de datos. Después envía la trama obtenida a su destino. Como el testigo ya no existe, las demás estaciones no pueden trasmitir. Cuando la trama enviada da toda la vuelta a la red, es captada otra vez por el emisor y éste introduce un nuevo testigo en la red. De esta forma, ya es posible que otra estación pueda emitir. Para baja carga de la red, este sistema es poco eficiente, pero para cargas altas, es un sistema muy eficiente y equitativo. Una desventaja seria es que se pierda el testigo, en cuyo caso toda la red se bloquearía. Los bits que se modifican en el anillo indican si la trama que acompaña al anillo ha llegado a su destino, si no ha llegado o si ha llegado pero no se ha copiado. Esta información de control es muy importante para el funcionamiento del sistema.

Algoritmo de Acceso al medio en IEEE 802.5: La trama consta de un campo de reserva de trama y un campo de prioridad de la propia trama, además de otros campos de control de errores y de los datos. Este estándar admite la posibilidad de utilizar prioridades. El algoritmo es:

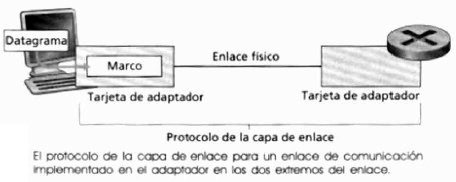
• Una estación que desee transmitir debe esperar un testigo con prioridad inferior a la suya propia.

• Si el emisor detecta una trama de datos, si su prioridad es superior a la de la reserva, pone su prioridad en un campo de reserva de la trama. Si lo recibido es una trama de testigo, si la prioridades mayor que la de la reserva y que la del propio testigo, pone su prioridad en el campo de reserva del testigo, eliminando a la que había.

• Cuando un emisor consigue el testigo, pone su prioridad en el campo de prioridad del testigo y pone a 0 el campo de reserva de testigo.

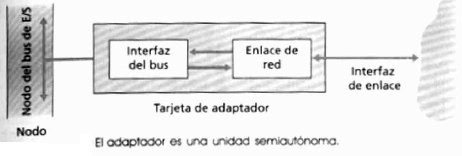
***Comunicación entre adaptadores***

Para un enlace de comunicación dado, el protocolo de la capa de enlace está, en su mayor parte, implementado en un adaptador. Un adaptador es una placa que contiene típicamente memoria RAM, chips DSP, una interfaz de bus del host, y una interfaz de enlace. Los adaptadores se conocen comúnmente como tarjetas de interfaz de red, o NIC. Como se ha visto en la figura:



La capa de red pasa un datagrama de la capa de red al adaptador que mantiene el enlace de comunicación. El adaptador encapsula el datagrama en un marco y transmite el marco en el enlace de comunicación. En el otro lado, el adaptador de recepción recibe el marco completo, extrae el datagrama de la capa de red, y lo pasa a la capa de red. Si el protocolo de la capa de enlace proporciona detección de errores, entonces es el adaptador de envío el que habilita los bits de detección de errores, y el adaptador de recepción el que realiza la comprobación de errores. Si el protocolo de la capa de enlace proporciona entrega fiable, los mecanismos para entrega fiable (por ejemplo números de secuencia, temporizadores o reconocimientos) están implementados en los adaptadores completamente. Si el protocolo de la capa de enlace proporciona acceso aleatorio, entonces el protocolo de acceso aleatorio está implementado completamente en los adaptadores.

Un adaptador es una unidad semiautónoma. Por ejemplo, un adaptador puede recibir un marco, determinar si el marco tiene error, y desechar el marco sin notificarlo a su nodo padre. Cuando recibe un marco, un adaptador interrumpirá a su nodo padre sólo si quiere pasar un datagrama de la capa de red hacia arriba de la pila de protocolos. De forma parecida, cuando un nodo pasa un datagrama hacia abajo en la pila de protocolos a un adaptador, el nodo delega totalmente en el adaptador la tarea de transmitir el datagrama sobre el enlace. Si bien un adaptador es semiautomático, no es una unidad totalmente autónoma, está alojado normalmente en la misma caja física que el resto del nodo, comparte la alimentación y los buses con el resto del nodo, y está en el fondo bajo el control del nodo.

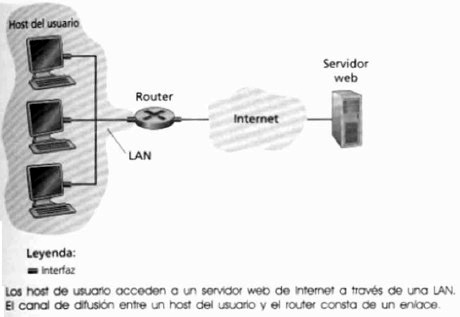


Como se ve en la figura, los componentes del adaptador, son la interfaz del bus y la del enlace. La interfaz del bus es responsable de comunicar con el nodo padre del adaptador. Transfiere datos e información de control entre el adaptador y el nodo padre. La interfaz del enlace es responsable de implementar el protocolo de la capa de enlace (ejemplo Ethernet). Además de enmarcar y desenmarcar datagramas, puede proporcionar detección de errores, acceso aleatorio, y otras funciones de la capa de enlace. También incluye la circuitería de transmisión y recepción.

***Redes de área local (LAN)***

Los protocolos de acceso múltiple se utilizan conjuntamente con muchos tipos diferentes de canales de difusión. Se han utilizado para canales de satélite y sin cable, donde los nodos transmiten sobre un espectro de frecuencia habitual. Se utilizan actualmente en el canal desde el usuario a la red para acceso de cable a Internet, y se utilizan ampliamente en redes de área local (LAN).

Recuerde que una LAN es una red de computadores que está concentrada en un área geográfica, como un edificio o un campus universitario. Para este tipo de acceso a Internet, el host del usuario está en un nodo de la LAN, y la LAN proporciona acceso a Internet a través de un router, como se ve en la figura.



La LAN es un simple enlace entre el host de cada usuario y el router; utiliza, por tanto, un protocolo de la capa de enlace, parte del cual es un protocolo de acceso múltiple.

En la década de 1980 y en los primeros años de la década de 1990, eran populares dos clases de tecnologías LAN:

* Paso de testigo:
  + Token ring (conocida también como IEEE 802.5)
  + FDDI (conocida también como interfaz de datos distribuidos de fibra), se diseñó para redes de área local más amplias geográficamente, como las llamadas redes de área metropolitana (MAN). Para redes amplias geográficamente (WAN) (expandidas sobre varios kilómetros) es poco eficiente dejar que un marco se propague hacia el nodo emisor una vez que ha pasado el nodo destino. En FDDI el nodo destino tiene que eliminar el marco del anillo. (Estrictamente hablando, FDDI no es un canal de difusión puro, ya que cada nodo no recibe cada marco transmitido).
* Redes de área local Ethernet (conocidas también como redes de área local 802.3, que se basan en acceso aleatorio.

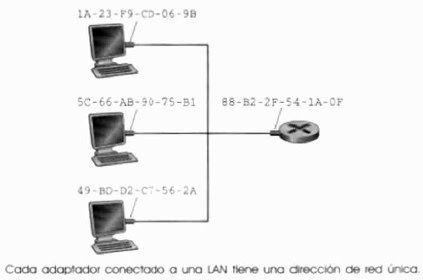
***Direcciones LAN y ARP***

Los nodos de las redes de área local envían marcos a los otros nodos sobre un canal de difusión. Esto significa que cuando un nodo de una LAN transmite un marco, todos los nodos conectados a la red reciben el marco. Pero normalmente un nodo de una LAN no quiere enviar un marco a todos los demás nodos de la LAN, sino que se lo quiere enviar a un nodo particular de la red. Para proporcionar esta funcionalidad, los nodos de la LAN necesitan ser capaces de direccionar a cualquier otro cuando envían marcos: es decir, los nodos necesitan direcciones de red de área local, y el marco de la capa de enlace necesita un campo que contenga tal dirección de destino. De esta forma, cuando un nodo recibe un marco, puede determinar si el marco estaba dirigido a él o a algún otro nodo de la LAN:

* Si la dirección de destino del marco coincide con la dirección de LAN del nodo receptor, entonces el nodo extrae el datagrama de la capa de enlace y pasa el datagrama hacia arriba en la pila de protocolos.
* Si la dirección de destino no concuerda con la dirección del nodo receptor, el nodo simplemente elimina el marco.

Direcciones LAN

En realidad, no es el nodo el que tiene una dirección LAN, sino el adaptador del nodo. Esto se muestra en la figura.



Una dirección LAN se llama también de varias formas, como dirección física, dirección Ethernet, o dirección MAC (control de acceso al medio). Para la mayoría de las redes de área local (incluyendo Ethernet y las redes de paso de testigo), la dirección LAN tiene una longitud de 6 bytes (6 Bytes \* 8 bits), lo que da 2^48 direcciones de LAN posibles. Como se vio en la figura, estas direcciones de seis bytes se expresan normalmente en notación hexadecimal, con cada byte de la dirección expresado como un par de número hexadecimales. La dirección LAN de un adaptador es permanente (cuando se fabrica un adaptador, se imprime una dirección LAN en la ROM de éste).

Una propiedad interesante de las direcciones de LAN es que no hay dos adaptadores con la misma dirección. Esto puede parecer sorprendente, dado que los adaptadores están fabricados en muchos países diferentes por muchas empresas diferentes.

Como hemos descrito al comienzo de esta sección, cuando un adaptador quiere enviar un marco a algún adaptador de destino en la misma LAN, el adaptador emisor inserta la dirección del adaptador de destino en el marco. Cuando el adaptador de destino recibe el marco, extrae el datagrama encerrado, y pasa el datagrama hacia arriba en la pila de protocolos. El resto de los adaptadores de la LAN también recibe el marco. Sin embargo, estos otros adaptadores descartan el marco sin pasar el datagrama de la capa de red hacia arriba de la pila de protocolos. Por tanto, estos otros adaptadores no tienen que interrumpir a su nodo padre cuando reciben datagramas destinados a otros nodos. Sin embargo, a veces un adaptador emisor quiere que todos los nodos restantes de la LAN reciban y procesen el marco que se va a enviar. En este caso, el adaptador emisor inserta una dirección de difusión de LAN en el campo de la dirección de destino del marco. Para redes de área local que usan direcciones de 6 bytes (como las de redes de área local Ethernet y de paso de testigo), la dirección de difusión es una cadena de 48 unos consecutivos (es decir, FF-FF-FF-FF-FF-FF en notación hexadecimal).

***Protocolo de resolución de direcciones (ARP)***

Como hay direcciones de la capa de red (por ejemplo, direcciones IP Internet) y direcciones de la capa de enlace (direcciones de LAN), hay una necesidad de traducción entre ellas. En Internet, éste es el trabajo del protocolo de resolución de direcciones (ARP) [RFC 826]. Cada host y cada router Internet de una LAN tienen un módulo ARP.

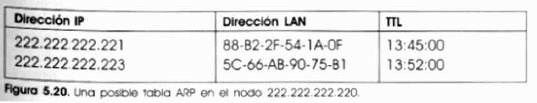
***Para motivar ARP, consideremos la red:***



En este ejemplo sencillo, cada nodo tiene una única dirección IP, y cada adaptador de nodo tiene una dirección LAN. Como es usual, las direcciones IP se muestran en una notación decimal puntuado, y las direcciones LAN se muestran en dirección hexadecimal. Ahora, supongamos que el nodo con la dirección IP 222.222.222.220 quiere enviar un datagrama IP al nodo 222.222.222.222. Para llevar a cabo esta tarea, el nodo emisor debe proporcionara su adaptador no sólo el datagrama IP, sino también la dirección LAN del nodo 222.222.222.222. Cuando se pasa el datagrama IP y la dirección LAN, el adaptador del nodo emisor construirá un marco de datos de la capa de enlace que contenga la dirección de LAN del nodo receptor, y enviará el marco a la LAN. Pero, ¿cómo conseguir que el nodo emisor determine la dirección LAN del nodo con dirección IP 222.222.222.222? Lo hace proporcionando a su módulo ARP la dirección IP 222.222.222.222, ARP responde entonces con la dirección LAN correspondiente, a saber, 49-BD-D2-C7-56-2A.

Vemos, por tanto, que ARP resuelve una dirección IP para una dirección LAN. En muchos sentidos, esto equivale a lo que ocurre con DNS, que resuelve los nombres de host para las direcciones IP. Sin embargo, una diferencia importante entre los dos resolvedores es que DNS resuelve los nombres de los host en cualquier parte de Internet, mientras que ARP resuelve direcciones IP sólo para los nodos de la misma LAN. Si un nodo en California intentara utilizar ARP para resolver las direcciones IP para un nodo en Mississipi, ARP devolvería un error.

Ahora que hemos explicado lo que hace ARP, vamos a ver cómo trabaja. El módulo ARP de cada nodo tiene una tabla en su RAM, llamada tabla ARP. Esta tabla contiene la traducción de las direcciones IP a direcciones LAN.

*Muestra cómo aparecería una tabla ARP en el nodo 222.222.222.220.*

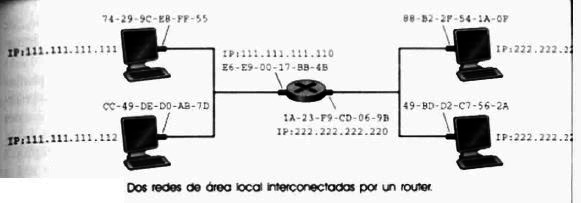
Para cada traducción de direcciones, la tabla contiene también una entrada de tiempo de vida (TTL), que indica cuándo la entrada será borrada de la tabla. Observe que la tabla no contiene necesariamente una entrada para cada nodo de la LAN; algunos nodos podrían tener entradas que hubieran expirado el tiempo, mientras que otros nodos podrían no haber sido introducidas nunca en la tabla. Un tiempo de expiración típico para una entrada son 20 minutos desde que se coloca una entrada en una tabla ARP. Ahora suponemos que el nodo 222.222.222.220 quiere enviar un datagrama que está direccionado IP a otro nodo en esa LAN. El nodo emisor necesita obtener la dirección LAN del nodo destino a partir de la dirección IP del nodo. Esta tarea es fácil si la tabla ARP del nodo emisor tiene una entrada para el nodo de destino. Pero, ¿qué pasa si la tabla ARP no tiene actualmente una entrada para el nodo de destino? En particular, supongamos que el nodo 222.222.222.220 quiere enviar un datagrama del nodo 222.222.222.222. En este caso, el nodo emisor utiliza el protocolo ARP para resolver la dirección. Primero, el nodo emisor construye un paquete especial, llamado paquete ARP. Un paquete ARP tiene varios campos, entre ellos las direcciones IP y LAN de envío y recepción. Tanto los paquetes de consulta ARP como los de respuesta tienen el mismo formato. El propósito del paquete de consulta ARP es consultar a los demás nodos de la LAN para determinar las direcciones LAN que corresponden a la dirección IP que se está resolviendo.

Volviendo a nuestro ejemplo, el nodo 222.222.222.220 pasa un paquete de consulta ARP al adaptador con una indicación para que el adaptador envíe el paquete a la dirección de difusión de LAN FF-FF-FF-FF-FF-FF. El adaptador encapsula el paquete ARP en un marco de enlace de datos, utiliza las direcciones de difusión para la dirección de destino del marco, y transmite el marco en la LAN. El marco que contiene la consulta ARP es recibido por los demás adaptadores de la LAN, y (debido a la dirección de difusión) cada adaptador pasa el paquete ARP con el marco hacia su nodo anfitrión. Cada nodo realiza una comprobación para ver si su dirección IP concuerda con la dirección IP de destino en el paquete ARP. El único nodo con una concordancia reenvía al nodo consultante un paquete de respuesta ARP con la traducción deseada. El nodo consultante (222.222.222.220) puede actualizar entonces su tabla ARP y enviar su datagrama IP.

Hay un par de cosas interesantes que señalar sobre el protocolo ARP. Primero, el mensaje de consulta ARP se envía con un marco de difusión, mientras que el mensaje de respuesta ARP se envía en un marco estándar. Segundo, ARP es de tipo conectar y funcionar; es decir, una tabla ARP de nodo se construye automáticamente (no tiene que ser configurada por un administrador del sistema). Y por último, si se desconecta un nodo de la LAN, su entrada será borrada finalmente de la tabla.

***Envío de un datagrama a un nodo de fuera de la LAN***

Ahora debería estar claro cómo opera ARP cuando un nodo quiere enviar un datagrama a otro nodo de la misma LAN (de la misma red IP). Pero veamos ahora una situación más complicada: un nodo de una red quiere enviar un datagrama de la capa de red a un nodo fuera de la LAN (de otra red IP). Vamos a discutir este tema en el contexto de la figura, que muestra una red sencilla formada por dos redes de área local interconectadas por un router:



Hay varias cosas interesantes que señalar. Primero, hay dos tipos de nodos: host y routers. Cada host tiene exactamente una dirección IP y un adaptador. Pero, un router tiene una dirección IP para cada una de sus interfaces. Cada interfaz del router tiene también su propio módulo ARP (en el router) y su propio adaptador. Como el router de la figura tiene dos interfaces, tiene dos direcciones IP, dos módulos ARP y dos adaptadores. Naturalmente, cada adaptador a la red tiene su propia dirección LAN.

Tenga en cuenta también que todas las interfaces conectadas a la LAN 1 tienen direcciones de la forma 111.111.111.xxx, y todas las conectadas a la LAN 2 tienen la forma 222.222.222.xxx. Por tanto, en este ejemplo, los tres primeros bytes de la dirección IP especifican la red, mientras que el último byte especifica la interfaz específica de la red. En la notación CIDR, la LAN 1 tiene las direcciones de red 111.111.111.000/24 y la LAN 2 tiene las direcciones de red 222.222.222.000/24.

Supongamos ahora que el host 111.111.111.111 quiere enviar un datagrama IP al host 222.222.222.222. El host emisor pasa el datagrama a su adaptador, como es habitual. Pero el host emisor debe indicar también a su adaptador una dirección LAN de destino apropiada. ¿Qué dirección LAN debería utilizar el adaptador? Se podría aventurar que la dirección LAN apropiada es la del adaptador para el host 222.222.222.222, es decir, 49-BD-D2-C7-56-2A. Esta suposición es, sin embargo, errónea. Si el adaptador de envío fuera a utilizar esa dirección LAN, ninguno de los adaptadores de la LAN 1 se molestaría en pasar el datagrama IP hacia arriba a su capa de red, puesto que la dirección de destino del marco no concordaría con la dirección LAN de ningún adaptador de la LAN 1. El datagrama seria desechado.

Si observamos cuidadosamente la figura, veremos que con el fin de que un datagrama vaya desde 111.111.111.111 a un nodo de la LAN 2, el datagrama debe ser enviado inicialmente a la interfaz del router 111.111.111.110. La tabla de ruteo en el host 111.111.111.111 debería indicar que, para alcanzar el host 222.222.222.222, el datagrama debería ser enviado a la interfaz del router 111.111.111.110. Por tanto, la dirección LAN apropiada para el marco es la dirección del adaptador para la interfaz del router 111.111.111.110, es decir, E6-E9-00-17-BB-4B ¿Cómo adquiere el host emisor la dirección LAN de 111.111.111.110? ¡Utilizando ARP!. Una vez que el adaptador del emisor tiene esta dirección LAN, crea un marco y lo envía a la LAN 1. El adaptador del router de la LAN 1 ve que el marco de enlace de datos está dirigido a él, y pasa por tanto el marco a la capa de red del router. ¡Hurra! ¡El datagrama IP se ha movido con éxito desde el host fuente al destino! Pero no hemos acabado. Todavía tenemos que mover el datagrama desde el router hasta el destino. El router tiene que determinar ahora la interfaz correcta a la que el datagrama será difundido. Esto se realiza consultando una tabla de ruteo en el router. La tabla de ruteo le dice al router que el datagrama se está enviando a través de la interfaz del router 222.222.222.220. Esta interfaz pasa entonces el datagrama a su adaptador, que encapsula el datagrama en un nuevo marco y envía el marco a la LAN 2. En este instante, la dirección de destino LAN es efectivamente la dirección LAN del destino definitivo. ¿Y cómo obtiene el router esta dirección LAN? ¡De ARP, naturalmente!. ARP para Ethernet está definido en RFC 826.

***Ethernet***

Se podría decir que Ethernet ha sido para las redes de área local lo que ha sido Internet para las redes globales.

Hay muchas razones para el éxito de Ethernet:

* **Primero**, fue la primera LAN de alta velocidad desplegada ampliamente. Como fue desplegada muy pronto. los administradores de redes llegaron a familiarizarse íntimamente con Ethernet (sus maravillas y sus rarezas), y fueron reacios a cambiar a otras tecnologías LAN cuando llegaron a escena.
* **Segundo**, token ring, FDDI y ATM eran más complejas y costosas que Ethernet, lo que desanimaba más a los administradores de red a cambiar.
* **Tercero**, la razón más determinante para cambiar a otra tecnología LAN (como FDDI o ATM) era normalmente la elevada tasa de datos de la nueva tecnología; sin embargo. Ethernet siempre se defendió, produciendo versiones que trabajaban a tasas de datos iguales o más elevadas. Ethernet conmutado, que se introdujo también en los primeros años de la década de 1990, aumentó sus tasas efectivas de datos.
* **Por último**, como Ethernet ha sido tan popular, el hardware para Ethernet (particularmente, adaptadores, hubs y switches) ha llegado a ser cómodo y considerablemente barato. El bajo coste se debe también al hecho de que el protocolo de acceso múltiple de Ethernet, CSMA/CD, está descentralizado completamente, lo que ha contribuido a un diseño sencillo.

La red de área local Ethernet original, como se ha mostrado en la figura, fue inventada a mediados de la década de 1970 por Bob Metcalfe y David Boggs.

***Estructura del marco Ethernet***

Dado que hay muchas tecnologías Ethernet diferentes actualmente en el mercado, ¿qué tienen en común?; ¿qué les une a todas bajo un nombre común? Lo primero y principal es la estructura del marco Ethernet. Todas las tecnologías Ethernet (utilicen cable coaxial o cable de cobre, funcionen a 10 Mbps, 100 Mbps, o 1Gbps) utilizan la misma estructura de marco.



*Marco Ethernet*.

Para entender Ethernet, consideremos el envío de un datagrama IP desde un host a otro, con ambos host en la misma LAN Ethernet. Consideremos que el adaptador de envío, el adaptador A, tiene la dirección física AA-AA-AA-AA-AA-AA, y el adaptador receptor, el adaptador B, tiene la dirección física BB-BB-BB-BB-BB-BB. El adaptador de envío encapsula el datagrama IP en el marco Ethernet y pasa el marco a la capa física. El adaptador receptor recibe el marco desde la capa física, extrae el datagrama IP, y pasa dicho datagrama a la capa de red. En este contexto, vamos a examinar los seis campo? del marco Ethernet:

* **Campo de datos** (46 a 1.500 bytes), lleva el datagrama IP. La unidad de transferencia máxima (MTU) de Ethernet es de 1.500 bytes. Esto significa que si el datagrama IP supera los 1.500 bytes, entonces el host debe fragmentar el datagrama. El tamaño mínimo del campo de datos es de 46 bytes. Esto significa que si el datagrama es menor de 46 bytes, el campo de datos tiene que ser rellenado hasta los 46 bytes. Cuando se utiliza el rellenado, los datos pasados a la capa de red contienen el relleno, así como un datagrama IP. La capa de red utiliza el campo longitud de la cabecera del datagrama IP para eliminar el relleno.
* **Dirección de destino** (6 bytes), contiene la dirección LAN del adaptador del destino, BB-BB-BB-BB-BB-BB. Cuando el adaptador B recibe un marco Ethernet con una dirección de destino distinta de su propia dirección física, BB-BB-BB-BB-BB-BB, o de la dirección de difusión de la LAN, lo desecha. Si no, pasa el contenido del campo de datos a la capa de red.
* **Dirección de origen** (6 bytes), contiene las direcciones LAN del adaptador que transmite el marco a la red, AA-AA-AA-AA-AA-AA.
* **Campo de tipo** (2 bytes), permite multiplexar los protocolos de la capa de red. Para comprender esta idea, necesitamos tener en cuenta que los host pueden utilizar otros protocolos distintos de la capa de red además de IP. De hecho, un host dado puede soportar múltiples protocolos de la capa de red que utiliza protocolos diferentes para aplicaciones diferentes. Por esta razón, cuando el marco Ethernet llega al adaptador B, éste necesita saber qué protocolo de la capa de red debe pasar (es decir, demultiplexar) el contenido del campo de datos. IP y otros protocolos tienen todos su propio número de tipo estandarizado. Además, el protocolo ARP tiene su propio número de tipo. Tenga en cuenta que el campo de tipo es análogo al campo de protocolo del datagrama de la capa de red y a los campos de número de puerto del segmento de la capa de transporte; todos estos campos sirven de pegamento entre un protocolo de una capa y otro protocolo de la capa superior.
* **Comprobación de redundancia cíclico** (CRC) (4 bytes), el propósito del campo CRC es permitir al adaptador de recepción, el adaptador B, detectar si se han introducido errores en el marco, es decir, si los bits del marco han sido cambiados. La causa de los errores de bit puede ser la atenuación de la intensidad de la señal y la energía electromagnética ambiental que se escapa por los cables y tarjetas de interfaz de Ethernet. La detección de errores se realiza de la forma siguiente. Cuando el host A construye el marco Ethernet, calcula el campo CRC, que se obtiene de una proyección de los otros bits del marco (excepto para los bits del preámbulo). Cuando el host B recibe el marco, aplica la misma proyección al marco y comprueba si el resultado de otra proyección es igual a lo que está en el campo CRC. Esta operación en el host receptor se llama comprobación CRC. Si la comprobación CRC falla (es decir, si el resultado de la proyección no es igual al contenido del campo CRC), entonces el host B sabe que hay un error en el marco.
* **Preámbul**o (8 bytes), cada uno de los primeros 7 bytes del preámbulo tiene un valor de 10101010; el último byte es 10101011. Los primeros 7 bytes del preámbulo sirven para despertar a los adaptadores receptores y para sincronizar sus relojes con el reloj del emisor. ¿Por qué deben realizar la sincronización los relojes? Tenga en cuenta que el adaptador A intenta transmitir el marco a 10 Mbps, 100 Mbps, o 1 Gbps, dependiendo del tipo de la LAN Ethernet. Sin embargo, como no hay nada absolutamente perfecto, el adaptador A no transmitirá el marco exactamente a la tasa destino; habrá siempre algún desplazamiento de la tasa objetivo, un desplazamiento que no es conocido a priori por los otros adaptadores de la LAN. Un adaptador receptor puede bloquear un reloj del adaptador A simplemente bloqueando los bits en los primeros 7 bytes del preámbulo. Los últimos dos bits del octavo byte del preámbulo (los primeros dos unos consecutivos) alertan al adaptador B de que el material importante está a punto de llegar. Cuando el host B ve los dos unos consecutivos, sabe que los 6 siguientes bytes son la dirección de destino. Un adaptador puede decir cuándo un marco finaliza simplemente detectando la ausencia de corriente.

***Un servicio sin conexión y no fiable***

Todas las tecnologías de Ethernet proporcionan servicio sin conexión en la capa de red. Es decir, cuando el adaptador A quiere enviar un datagrama al adaptador B, encapsula el datagrama en un marco Ethernet y envía el marco en la LAN, sin acuerdo inicial con el adaptador B. Este servicio sin conexión de la capa 2 es análogo al servicio de datagramas de la capa 3 de IP, y al servicio sin conexión de la capa 4 de UDP. Incluso si no ha habido colisión, un marco recibido puede contener errores de bit debidos al ruido del canal de comunicación. Todas las tecnologías de Ethernet proporcionan un servicio no fiable a la capa de red. Específicamente, cuando el adaptador B recibe un marco desde el adaptador A, ejecuta sobre el marco una comprobación CRC, pero no envía un acuse de recibo cuando un marco pasa la comprobación CRC (ni envía una confirmación negativa cuando un marco no pasa la comprobación CRC). El adaptador A no tiene ni la más ligera idea de si el marco transmitido ha pasado la comprobación CRC. Cuando un marco no pasa dicha comprobación, el adaptador B lo desecha. Esta falta de transporte fiable (en la capa enlace) ayuda a que Ethernet sea simple y barato. Pero también significa que la secuencia de datagramas que se pasa a la capa de red puede tener huecos.

Si hay huecos debido a los marcos Ethernet desechados, ¿debe la aplicación en el host B ver también los huecos? Como hemos aprendido en el Capítulo 3, esto depende únicamente de si la aplicación está utilizando UDP o TCP. Si la aplicación está utilizando UDP, entonces tendrá huecos en los datos. Por otro lado, si la aplicación está utilizando TCP, entonces TCP no confirmará los datos desechándolos, obligando a que TCP en el host A retransmita. Observe que cuando TCP retransmite datos, los datos vuelven eventualmente al adaptador Ethernet en el que fueron desechados. Por tanto, en este sentido, Ethernet puede retransmitir datos. Pero debemos tener en cuenta que Ethernet no sabe lo que está retransmitiendo, Ethernet piensa que está recibiendo un nuevo datagrama con nuevos datos, aunque el datagrama contenga los datos que ya se habían transmitido al menos una vez.

***Redes sin cable: IEEE 802.11b***

Las redes de área local sin cable están disfrutando actualmente de un despliegue rápido en los departamentos de las universidades, oficinas de negocios, cafés, hospitales y hogares. Por ejemplo, los usuarios residenciales están desplegando tecnología LAN sin cable barata en sus hogares, permitiendo que múltiples usuarios de la casa accedan simultáneamente a Internet mientras se mueven por la misma. Actualmente, hay varios estándares y tecnologías para redes de área local sin cable. Pero el estándar que está disfrutando de un despliegue más amplio es el IEEE 802.11b (también conocido como Ethernet sin cable, o Wi-Fi, o "wireless local área network" (WLAN)), que transmite en un espectro de radio que no precisa autorización a 2.4 GHz, y proporciona acceso a Ethernet sin cable a 11 Mbps.

El estándar 802.11b define la capa física y la capa de control de acceso al medio (MAC) para una red de área local sin cable. La capa física utiliza Direct Sequence Spread Spectrum (DSSS), que codifica cada bit en un patrón de bit, llamado un código chipping. Esta técnica es semejante a la utilizada en CDMA, excepto que ahora todos los host móviles (y las estaciones base) utilizan el mismo código chipping. Como todos utilizan el mismo código, DSSS no es un protocolo de acceso múltiple; es decir, no intenta coordinar el acceso al canal desde múltiples host, sino que es un mecanismo de la capa física que propaga la energía en una señal sobre un rango de frecuencia amplio, mejorando por ello la capacidad del receptor para recuperar los bits originales transmitidos.

IEEE 802.11b pertenece a una familia de protocolos LAN sin cable llamados colectivamente "IEEE 802.11". Otros miembros de esta familia son: 802.11a, que funciona entre rangos de 5 GHz y 6 GHz utiliza multiplexación por división de frecuencia ortogonal (OFDM) en lugar de DSSS, y proporciona velocidades de datos de hasta 54 Mbps; y 802.llg , que funciona a 2.4 GHz (como 802.11b) y proporciona tasa de datos de hasta 54 Mbps (como 802.11a). Todos los estándares 802.11 tienen la misma arquitectura y utilizan el mismo protocolo MAC.

***PPP: El protocolo punto a punto***

La mayor parte de nuestra discusión sobre protocolos de enlace de datos se ha enfocado especialmente en protocolos para canales de difusión. En esta sección, tratamos un protocolo de enlace de datos para enlaces punto a punto: PPP, el protocolo punto a punto. Como PPP es típicamente el protocolo de elección para un enlace telefónico para host residenciales, es indudablemente uno de los protocolos de enlace de datos desplegados hoy más ampliamente.

Como implica su nombre, el protocolo punto a punto (PPP) [RFC 1661; RFC 2153] es un protocolo de la capa de enlace de datos que opera sobre un enlace punto a punto (un enlace que conecta directamente dos nodos, uno en cada lado del enlace). El enlace punto a punto sobre el que funciona PPP puede ser una línea de modem telefónico, un enlace SONET/SDH, una conexión X.25, o un circuito ISDN. Como hemos indicado anteriormente, PPP se ha convertido en el protocolo elegido para conectar a los usuarios de casa a sus ISP sobre una conexión de modem telefónico.

SE PUEDE VER UN MAYOR DESARROLLO DEL TEMA ENTRE LAS IMAGENES 496 Y 501 DEL LIBRO REFERENCIA DE LA CATEDRA.

***Modo de transferencia asíncrona (ATM)***

Los estándares para ATM fueron desarrollados inicialmente a mediados de la década de 1980. Para los que son demasiado jóvenes para recordar, en aquella época había predominantemente dos tipos de redes: redes de teléfonos, que eran (y aún lo son) utilizadas principalmente para llevar voz en tiempo real, y redes de datos, que se utilizaban fundamentalmente para transferir archivos de texto, soportar login remoto, y proporcionar correo electrónico. Había también redes privadas dedicadas disponibles para videoconferencia. Internet existía en esa época, pero poca gente estaba pensando en cómo utilizarla para transportar llamadas de teléfono, y todavía no se había oído hablar de la World Wide Web. Por consiguiente, era natural diseñar una tecnología de red que fuera apropiada para transportar audio y video en tiempo real, así como texto, correo electrónico y archivos de imagen. El modo de transferencia asíncrona (ATM) consiguió este objetivo. Dos comités de estándares, el foro ATM [ATM 2002] y la Unión Internacional de Telecomunicaciones [ITU 2002], desarrollaron estándares para redes de servicios digitales de banda ancha.

Los estándares ATM requirieron conmutación de paquetes con circuitos virtuales (llamados canales virtuales en la jerga ATM). Los estándares definen cómo interconectar las aplicaciones directamente con ATM, por lo que ATM proporciona una solución completa para aplicaciones distribuidas. En paralelo con el desarrollo de estándares ATM, las principales empresas del mundo hicieron inversiones significativas en investigación y desarrollo para ATM. Estas inversiones han conducido a un gran número de tecnologías ATM de prestaciones elevadas, como los switches ATM, que pueden conmutar terabits por segundo. En años recientes, la tecnología ATM ha sido desplegada muy agresivamente, tanto en redes de teléfono como en troncales Internet.

Aunque ATM ha sido desplegada en redes, ha tenido menos éxito extendiéndose de todas las formas a los PC de sobremesa y las estaciones de trabajo. Y ahora es cuestionable si ATM tendrá alguna vez una presencia significativa en los computadores de sobremesa. Ciertamente, mientras ATM se estaba elaborando en los comités de estándares y laboratorios de investigación a finales de la década de 1980 y en los primeros años de la de 1990, Internet y sus protocolos TCP/IP eran ya operativos y hacían progresos significativos:

* El paquete de protocolos TCP/IP integraba en todos los sistemas operativos más populares.
* Las empresas comenzaron a tramitar comercio (e-commerce) sobre Internet.
* El acceso a Internet desde las casas se abarató.
* Muchas aplicaciones maravillosas para computadores de escritorio se desarrollaron para redes TCP/IP, incluyendo la World Wide Web, el teléfono Internet y el flujo de vídeo interactivo. Miles de empresas están desarrollando actualmente nuevas aplicaciones y servicios para Internet.

Hoy vivimos en un mundo en el que la mayoría de las aplicaciones de red interactúan sólo con TCP IP. No obstante, los switches ATM pueden reenviar datos a tasas muy altas, y consecuentemente han sido desplegados en redes troncales de Internet, en las que la necesidad de transportar tráfico a tasas elevadas es más aguda. Cuando ATM está en la troncal Internet, TCP/IP funciona por encima de ATM y se ve una red completa ATM, que puede abarcar un continente, como una red grande de la capa de enlace. En otras palabras, aunque ATM no ha llegado a mantenerse como una solución proceso a proceso (ni tampoco como una solución conmutador a computador), ha encontrado un nicho en el nivel de enlace en panes del troncal Internet; esto se conoce como IP sobre ATM.