Resumen

Redes de Comunicaciones (TB067) - FIUBA
Martin Klöckner - mklockner@fi.uba.ar

Introducción

Una **red** es un sistema de nodos interconectados mediante enlaces de comunicación, que utilizan protocolos para el intercambio de datos. La conexión puede darse a través de un único cable trenzado entre los extremos o a través de redes mas grandes, por ejemplo internet, que es una red de redes.

Un **protocolo** es un conjunto de reglas o convenciones utilizadas para controlar la transferencia de datos en una red.

Los enlaces físicos vinculan transmisor con receptor y la información se propaga a través de medios, estos medios pueden ser **guiados** o **no guiados**. Los medios guiados son aquellos que propagan las señales por medios sólidos como cables de cobre, mientras que los medios no guiados son aquellos que propagan las señales a través del espacio libre, haciendo uso de antenas, por ejemplo microondas.

Existen dos métodos fundamentales para transportar información a través de redes, la **conmutación de circuitos** y la **conmutación de paquetes**. En la conmutación de circuitos se establece un camino dedicado entre emisor y receptor antes de comenzar la transmisión, y ese camino permanece reservado durante toda la comunicación, un ejemplo de este método de comunicación es la red telefónica tradicional. En la conmutación de paquetes los datos se dividen en paquetes que se envían de forma independiente a través de la red sin haber reservado recursos previamente, estos paquetes comparten dinámicamente los recursos de la red, por ejemplo internet.

En la conmutación de paquetes los nodos intermedios típicamente utilizan un método de transmisión llamado **Store-and-Forward**, en el cual el nodo no retransmite la unidad de datos (trama o paquete) hasta que no disponga de la totalidad de los bits correspondientes a esa unidad. Este nodo intermedio almacena temporalmente la unidad de datos incompleta en un buffer hasta no recibirla por completo. Puede ocurrir que la tasa de arribo de unidades de datos exceda la tasa de transmisión (esto puede ocurrir por ejemplo si se tienen enlaces con diferentes características) en cuyo caso, si el nodo dispone de memoria libre se encola, produciendo un retardo de encolamiento, o si el nodo no dispone de memoria libre se descarta esa unidad, provocando una perdida de datos. Además del retardo de encolamiento existen otros retardos que caracterizan lo que se denomina **retardo extremo a extremo**, que es el tiempo total que tarda un paquete en ir desde el origen hasta el destino. Los retardos son: el retardo de procesamiento, el retardo de propagación y el retardo de transmisión. El retardo de procesamiento es el tiempo que se tarda un nodo en examinar el encabezado del paquete. El retardo de transmisión es el tiempo necesario para poner todos los bits del paquete en el enlace y resulta del cociente entre el tamaño del paquete L (en bits) y la tasa de transmisión del enlace R (en bits por segundo o bps). El retardo de propagación es el tiempo que tarda la señal en viajar físicamente por el medio, resulta del cociente entre la distancia del enlace d (en metros) y la velocidad de propagación del medio v_{prop} (en metros por segundo).

El **throughput** es la tasa a la que se envían los bits de la fuente a destino, el throughput extremo a extremo queda determinado por el enlace de menor tasa de transmisión R (en bits por segundo).

Un modelo de comunicaciones organiza las funciones necesarias para la comunicación de sistemas en una red en capas, de modo que cada capa cumple un rol específico. Los modelos tradicionales son el modelo TCP/IP (por los protocolos utilizados) y el modelo OSI (por Open Standard Institute) los cuales se muestran en las tablas 1a y 1b, respectivamente.

Aplicación
Transporte
Red
Enlace
Física

(a) Modelo TCP/IP

Aplicación
Presentación
Sesión
Transporte
Red
Enlace
Física

(b) Modelo OSI

Tabla 1: Modelos de comunicaciones

Nivel de aplicación

El nivel de aplicación es el mas alto en cuanto a nivel de abstracción, permite que las aplicaciones de los usuarios se comunique a través de la red, ocultando los detalles de transporte y de la red subyacente, por ejemplo aplicaciones de transferencia de archivos (FTP), de correo electrónico (SMTP), de resolución de direcciones IP (DNS) o servidores Web (HTTP), entre otras. El software del nivel de aplicación solo se ejecuta en los dispositivos terminales, ya que en los dispositivos intermedios, como son los switches y routers, solo se ejecuta software del nivel de red y enlace.

Existen dos paradigmas principales de la comunicación entre terminales: el paradigma cliente-servidor y el peer-to-peer (abreviado P2P). El paradigma cliente-servidor involucra la comunicación entre: un cliente, que realiza una petición por un servicio, y un servidor, que siempre debe estar disponible para responde a esa petición. En el paradigma peer-to-peer los dispositivos de una red intercambian recursos entre sí, compartiendo carga, control y datos, no hay un servidor siempre disponible para satisfacer las peticiones si no que cada nodo ofrece y solicita recursos.

Un proceso (o aplicación si se quiere) ejecutándose en una terminal, envía y recibe mensajes de la red a través de una interfaz de software denominada **socket**. Un socket es la interfaz entre la capa de aplicación y la capa de transporte de un host. El socket se compone de la dirección IP del dispositivo, el protocolo de transporte y el numero de **puerto**, el puerto identifica un servicio o aplicación especifica dentro del dispositivo, por convención se asignan diferentes números de puerto a diferentes aplicaciones [1] , siendo las más comunes el puerto 80 para HTTP, el puerto 443 para HTTPS, el puerto 22 para SSH, etc. A continuación se muestran ejemplos de sockets:

```
TCP; 157.92.49.38; 80
TCP; 122.36.99.208; 9887
```

Una **sesión** es la conexión de sockets entre extremos, sea punto-a.punto o cliente-servidor. Es un canal de comunicación estable y temporal entre dos dispositivos y se compone de camps correspondiente al nivel de transporte y red, en primer lugar el protocolo de transporte, luego la dirección IP y el numero de puerto del cliente y la dirección IP y el numero de puerto del servidor. Por ejemplo:

```
(UDP; 208.67.22.22; 23; 10.99.1.33;
23054)
```

WWW

La World Wide Web es un conjunto de protocolos y aplicaciones que se utilizan para transferir recursos sobre internet. Estos recursos pueden ser documentos HTML, texto plano, imágenes, entre otros, y se obtienen mediante enlaces (hipervínculos) accesibles mediante navegadores web, que utilizan los protocolos HTTP/HTTPS.

Un hipervínculo es una referencia activa que utiliza una URL (Uniform Resource Locator) para localizar el recurso al que apunta. Una URL es un identificador que indica dónde está un recurso y cómo acceder a él. Por ejemplo en la URL `http://www.ejemplo.com/img/meme.jpeg` el nombre de host es `www.ejemplo.com` mientras que la ruta es `/img/meme.jpeg`

HTTP

El protocolo HTTP (HyperText Transfer Protocol) o su version segura HTTPS (HyperText Transfer Protocol Secure) definen la estructura de los mensajes entre aplicaciones del cliente y el servidor en la Web. HTTP/HTTPS utiliza TCP como protocolo de transporte (por lo menos hasta la version HTTP/2) y se asigna por convención el puerto 80. Dado que se utiliza TCP como protocolo de transporte previo a realizar peticiones a un servidor el cliente HTTP debe establecer una conexión TCP con el servidor, una vez que se ha establecido la conexión entre el cliente y el servidor, a nivel de aplicación, se comunican mediante sockets.

El servidor envía los archivos solicitados a los clientes sin almacenar ninguna información acerca del estado del cliente. Si un determinado cliente pide el mismo objeto dos veces en un espacio de tiempo de unos pocos segundos, el servidor reenvía el objeto, ya que ha olvidado por completo que ya lo había hecho anteriormente. Dado que un servidor HTTP no mantiene ninguna información

[1] List of TCP and UDP port numbers, https://en.wikipedia.org/w/index.php?title=List_of_TCP_and_UDP_port_numbers

acerca de los clientes, se dice que HTTP es un protocolo **sin memoria del estado**.

Las conexiones HTTP entre cliente-servidor pueden ser **persistentes** o **no persistentes**, cuando la conexión es persistente se mantiene la misma conexión TCP para multiples solicitudes y respuestas entre el cliente y el servidor, si la conexión es no persistente, se utilizan conexiones TCP separadas para una petición y para otra.

El tiempo de ida y vuelta (Round-Trip Time, RTT) es el tiempo que tarda un paquete pequeño en viajar desde el cliente al servidor y volver de nuevo al cliente. El RTT incluye los retardos de propagación de los paquetes, los retardos de cola en los routers y switches intermedios y los retardos de procesamiento de los paquetes.

Como se mencionó previamente el protocolo HTTP define una estructura que deben tener los mensajes de solicitud y respuesta, a continuación se muestra un mensaje de solicitud típico.

```
GET /index.html HTTP/1.1
Host: www.ejemplo.com
Connection: close
User-agent: Mozilla/5.0
Accept-language: es
```

Un mensaje de respuesta al mensaje de solicitud previo puede ser el que se muestra a continuación.

```
HTTP/1.1 200 OK
Date: Tue, 06 Feb 2026 12:30:00 GMT
Server: Apache/2.4.57
Content-Type: text/html
Content-Length: 137

<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <h1>Hola, mundo!</h1>
  </body>
</html>
```

HTTP/1.0 (1996)

Esta es una mejora de la primera version del protocolo (HTTP/0.9 de 1991) en esta version se introdujo el concepto de cabeceras HTTP, lo que permitió transportar distintos tipos de contenido además de HTML, indicar metadatos y devolver códigos de estado.

HTTP/1.1 (1997-1999)

En esta versión se mejoró significativamente la eficiencia al incorporar conexiones persistentes, permitiendo reutilizar una misma conexión TCP para varias solicitudes. También se añadió soporte para hosts virtuales, mejores mecanismos de caché y control de errores. Se introdujo el concepto de pipelining lo cual permite enviar varias solicitudes sin esperar respuestas, pero esto no tuvo mucho éxito debido a problemas de bloqueo.

HTTP/2 (2015)

En esta version se cambió completamente la forma de transmitir los datos, aunque la semántica sigue siendo la misma que en versiones anteriores. Se utiliza un formato binario y multiplexa múltiples flujos de datos sobre una sola conexión TCP, reduciendo la latencia y mejorando el rendimiento. Además, incorpora compresión de cabeceras y la posibilidad de envío proactivo de recursos por parte del servidor, aunque sigue afectado por el bloqueo de cabeza de línea propio de TCP.

HTTP/3 (2022)

Esta es la versión más reciente y funciona sobre QUIC en lugar de TCP, utilizando UDP como protocolo subyacente. Esto permite evitar el bloqueo de cabeza de línea entre flujos, reducir el tiempo de establecimiento de conexión y soportar la migración de conexiones cuando cambia la dirección IP del cliente.

Cookies

Como se menciono previamente un servidor HTTP no tiene memoria del estado de la conexión, sin embargo existe el concepto de **cookie** que permite a los sitios seguir la pista de los usuarios, almacenando un numero de identificador en la terminal del cliente e información relacionada a ese identificador en el servidor. El sistema de cookies utiliza cuatro componentes: una linea de cabecera de la cookie en el mensaje de solicitud HTTP, `Cookie: 1234`; una linea de cabecera en el mensaje de respuesta HTTP, `Set-cookie: 1234`; el archivo de cookies almacenado en el sistema terminal del

usuario y gestionado por el navegador; y una base de datos en el sitio-web.

Cache web

Un cache web o también denominado servidor **proxy** guarda temporalmente respuestas HTTP para poder servirlas directamente a los clientes cuando el recurso no ha cambiado.

Aunque el almacenamiento en caché puede reducir los tiempos de respuesta percibidos por el usuario, introduce un nuevo problema: la copia de un objeto que reside en la caché puede estar desactualizada. En otras palabras, el objeto almacenado en el servidor web puede haber sido modificado desde que la copia fue almacenada en la caché del cliente. HTTP dispone de un mecanismo que permite a la caché verificar que sus objetos están actualizados. Este mecanismo se denomina GET condicional. Un mensaje de solicitud HTTP se denomina también mensaje GET condicional si el mensaje de solicitud utiliza el método GET y ademan se incluye una línea de cabecera `If-Modified-Since`. Por ejemplo un servidor cache desea saber si su versión del archivo `/fruit/kiwi.jpeg` esta desactualizada con respecto al servidor web, por lo que envía el siguiente mensaje HTTP al servidor.

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
If-modified-since: Wed, 9 Sep 2015
09:23:24
```

El servidor puede responder con el archivo actualizado o con un mensaje HTTP con código 304 que no ha sido modificado, como se muestra a continuación.

```
HTTP/1.1 304 Not Modified
Date: Sat, 10 Oct 2015 15:39:29
Server: Apache/1.3.0 (Unix)
(cuerpo de entidad vacío)
```

SMTP

El protocolo simple de transferencia de correo (Simple Mail Transfer Protocol, o SMTP) es el principal protocolo de transferencia de correo de la capa de aplicación, utiliza el protocolo de transporte TCP y por convención se asigna el puerto 25.

A diferencia del protocolo HTTP, el cual es un protocolo principalmente de tipo **pull** (o extracción) en el cual alguien carga los datos a un servidor web y los usuarios mediante HTTP extraen la información cuando desean, SMTP es un protocolo principalmente de tipo **push** (o inserción), en este caso el servidor de correo emisor introduce el archivo en el servidor de correo receptor. La máquina que desea enviar el archivo inicia la conexión TCP.

Cuando una persona envía un mensaje de correo electrónico a otra, existe una cabecera que contiene información administrativa y antecede al cuerpo del mensaje, análogo a la cabecera de HTTP. Toda cabecera tiene que estar formada por una línea de cabecera `From:` y una línea de cabecera `To:`; también puede incluir una línea `Subject:`, así como otras líneas de cabecera opcionales. Por ejemplo la siguiente cabecera

```
From: alicia@crepes.fr
To: benito@hamburger.edu
Subject: Hello, World!
```

Mail User Agent

El agente de usuario de mail es la aplicación que un usuario utiliza para conectarse con el servidor de correo.

Protocolos de acceso de correo

Un destinatario que ejecuta un agente de usuario en su PC local obtiene sus mensajes, que se encuentran en un servidor de correo de su ISP por ejemplo, mediante un protocolo especial de acceso al correo que permita transferir los mensajes del servidor de correo a su PC local, hay que tener en cuenta que no se puede utilizar el protocolo SMTP ya que seria una operación de extracción (*pull*). Actualmente existen varios protocolos de extracción de correo como ser POP3 (Post Office Protocol Version 3, Protocolo de oficina de correos versión 3) el Protocolo de acceso de correo de Internet (IMAP, Internet Mail Access Protocol) y también se puede utilizar HTTP.

DNS

El sistema de nombres de dominio (Domain Name System, DNS) es un servicio de la capa de aplicación que se utiliza para hallar la dirección IP de

un host solicitado por un cliente. DNS utiliza el protocolo de transporte UDP y se asigna por convención el puerto 53.

Por ejemplo, un navegador solicita el URL `www.unaescuela.edu/index.html`, para que el host del usuario pueda enviar un mensaje de solicitud HTTP al servidor web `www.unaescuela.edu` debe obtener primero la dirección IP de `www.unaescuela.edu`, por lo que realiza los siguientes pasos:

1. La máquina del cliente ejecuta el lado cliente de la aplicación DNS.
2. El navegador extrae el nombre de host `www.unaescuela.edu` del URL y lo pasa al lado cliente de la aplicación DNS.
3. El cliente DNS envía una consulta que contiene el nombre de host a un servidor DNS.
4. El cliente DNS recibe como respuesta la dirección IP correspondiente al nombre del host.
5. Teniendo la dirección IP del servidor, el navegador puede iniciar una conexión TCP.

Los servidores DNS se organizan de forma jerárquica y se distribuyen alrededor de todo el mundo. Ningún servidor DNS dispone de todas las correspondencias de todos los hosts de Internet. En una primera aproximación, podemos decir que existen tres clases de servidores DNS: los servidores DNS raíz, los servidores DNS de dominio de nivel superior (Top-Level Domain, TLD) y los servidores DNS autoritativos. Un servidor DNS **autoritativo** mantiene los registros oficiales de una zona DNS, mientras que un servidor **no autoritativo** resuelve consultas usando caché o consultando a otros servidores.

Suponga que un cliente DNS desea determinar la dirección IP correspondiente al nombre de host `www.amazon.com`. En una primera aproximación tienen lugar los siguientes sucesos: primero, el cliente contacta con uno de los servidores raíz, el cual devuelve las direcciones IP para los servidores TLD del dominio de nivel superior `com`. A continuación, el cliente contacta con uno de estos servidores TLD, que devuelve la dirección IP de un servidor autoritativo para `amazon.com`. Por último, el cliente contacta con uno de los servidores autoritativos de `amazon.com`, el cual devuelve la dirección IP correspondiente al nombre de host `www.amazon.com`.

Las consultas DNS pueden ser **recursivas** o **iterativas**. Una consulta recursiva exige una respuesta final, por lo que de no tener una respuesta el servidor DNS consulta a otros servidores DNS; en una consulta DNS iterativa el servidor devuelve la mejor información disponible, de no tener una entrada para la dirección de host solicitada responde con una referencia a otro servidor DNS.

La resolución de la dirección IP para un host dado, agrega un retardo cuando se quiera establecer una conexión entre un cliente y un servidor, es por eso que los servidores DNS utilizan exhaustivamente el **almacenamiento en cache**. Cuando un servidor DNS local, es decir, no dispone de la entrada DNS para un host pedido y debe recurrir a otro servidor DNS, recibe una respuesta DNS de resolución para ese host, almacena la respuesta, de manera tal que si en un futuro otro cliente realiza una petición de resolución DNS para ese host el servidor local disponga de la respuesta inmediatamente. Dado que los hosts y las correspondencias entre nombres de host y direcciones IP no son permanentes, los servidores DNS descartan la información almacenada en caché pasado un cierto período de tiempo (normalmente, unos dos días).

Los servidores DNS almacenan la información en tablas o **registros de recursos (RR)**, que es básicamente una base de datos. Los mensajes de respuesta DNS pueden contener uno o mas registros de recursos. Los registros de recursos están formados por cuatro campos Nombre, Valor, Tipo y TTL y se organizan de la siguiente forma (conceptual, en la practica el formato es distinto):

```
(Nombre, Valor, Tipo, TTL)
```

El campo TTL es el tiempo de vida del registro de recurso; determina cuándo un recurso debería ser eliminado de una caché DNS. El significado de Nombre y Valor depende del campo Tipo, el cual puede ser A, AAAA, NS, CNAME o MX.

En los registros A y AAAA, el campo Nombre es una dirección de host y el campo Valor es una dirección IP, IP version 4 en el caso del tipo A e IP version 6 en el caso de tipo AAAA. Por ejemplo el siguiente registro tipo A (en el cual se omite el campo TTL):

```
(relay1.bar.foo.com, 145.37.93.126, A)
```

En los registros de tipo NS el campo Nombre es un dominio (como `ejemplo.com`) y el campo Valor es el nombre de host de un servidor DNS autorita-

tivo que sabe cómo obtener las direcciones IP de los hosts de ese dominio. Este registro se utiliza para enrutar las consultas DNS a lo largo de la cadena de consultas. A continuación se muestra un ejemplo de registro de tipo NS:

```
(foo.com, dns.foo.com, NS)
```

En los registros de tipo CNAME el campo Valor es un nombre de **host canónico**, esto es un identificador único para un host dado, y el campo Nombre es un alias que apunta a ese host. Por ejemplo un alias puede ser `ejemplo.com` que "apunta" al nombre canónico `relay1.bar.ejemplo.com`.

En los registros de tipo MX, el campo Valor es un nombre canónico de un servidor de correo con un alias dado por el campo Nombre. Por ejemplo el registro `(foo.com, mail.bar.foo.com, MX)` es un registro MX. Los registros MX permiten a los nombres de host de los servidores de correo tener alias simples.

ping

```
root@redes:~# ping -4 -c 2 google.com
PING google.com (172.217.28.14) 56(84)
bytes of data.
64 bytes from lcezea-af-in-f14.1e100.net
(172.217.28.14): icmp_seq=1 ttl=119 time
=4.11 ms
64 bytes from lcezea-af-in-f14.1e100.net
(172.217.28.14): icmp_seq=2 ttl=119 time
=3.96 ms

--- google.com ping statistics ---
2 packets transmitted, 2 received, 0%
packet loss, time 1001ms
rtt min/avg/max/mdev =
3.955/4.032/4.109/0.077 ms
```

traceroute

```
root@redes:~# traceroute example.com
traceroute to example.com (104.18.26.120)
, 30 hops max, 60 byte packets
 1 10.0.2.2 (10.0.2.2) 0.135 ms 0.077 ms
0.057 ms
 2 192.168.1.1 (192.168.1.1) 0.427 ms
0.595 ms 0.717 ms
 3 200.51.241.1 (200.51.241.1) 4.393 ms
4.440 ms 5.300 ms
 4 200.51.240.61 (200.51.240.61) 4.494 ms
 5.515 ms 213.140.39.119 (213.140.39.119)
 5.963 ms
 5 213.140.39.116 (213.140.39.116) 5.533
ms 5.573 ms 5.730 ms
 6 cloudflare-ae70-0-grtbueba1.net.
telefonicaglobalsolutions.com
(94.142.103.101) 7.419 ms 5.53.7.242
(5.53.7.242) 3.135 ms 4.082 ms
 7 198.41.228.7 (198.41.228.7) 17.473 ms
198.41.228.5 (198.41.228.5) 9.111 ms
cloudflare-ae70-0-grtbueba1.net.
telefonicaglobalsolutions.com
(94.142.103.101) 8.244 ms
 8 198.41.228.7 (198.41.228.7) 4.496 ms
4.543 ms 4.683 ms
 9 104.18.26.120 (104.18.26.120) 4.321 ms
 4.655 ms 4.674 ms
```

dig

Nivel de transporte

Un protocolo de la capa de transporte proporciona una comunicación lógica entre procesos de aplicación que se ejecutan en hosts diferentes. Por comunicación lógica se dice que, desde la perspectiva de la aplicación, es como si los hosts que ejecutan los procesos estuvieran conectados directamente; en realidad, los hosts pueden encontrarse en puntos opuestos del planeta, conectados mediante numerosos routers y a través de un amplio rango de tipos de enlace. Los procesos de aplicación utilizan la comunicación lógica proporcionada por la capa de transporte para enviarse mensajes entre sí, sin preocuparse por los detalles de la infraestructura física utilizada para transportar estos mensajes.

Los protocolos de la capa de transporte están implementados en los sistemas terminales, pero no en los routers de la red. En el lado emisor, la capa de transporte convierte los mensajes que recibe procedentes de un proceso de aplicación emisor en paquetes de la capa de transporte, conocidos como **segmentos**, esto se hace dividiendo los mensajes de la aplicación en fragmentos más pequeños y añadiendo una cabecera de la capa de transporte a cada fragmento, con el fin de crear el segmento de la capa de transporte. Luego, la

capa de transporte pasa el segmento a la capa de red del sistema terminal emisor, donde el segmento se encapsula dentro de un paquete de la capa de red (un **datagrama**) y se envía al destino. En el lado receptor, la capa de red extrae el segmento de la capa de transporte del datagrama y lo sube a la capa de transporte. A continuación, esta capa procesa el segmento recibido, poniendo los datos del segmento a disposición de la aplicación de recepción.

Para aplicaciones de red puede haber más de un protocolo de la capa de transporte disponible como el Protocolo de datagramas de usuario (User Datagram Protocol, **UDP**) y el Protocolo de control de transmisión (Transmission Control Protocol, **TCP**).

Un protocolo de la capa de transporte proporciona una comunicación lógica entre procesos que se ejecutan en hosts diferentes, un protocolo de la capa de red proporciona una comunicación lógica entre hosts

Multiplexación y demultiplexación

La multiplexación y la demultiplexación son las funciones fundamentales que permiten que la capa de transporte extienda el servicio de entrega de datos de host a host a un servicio de entrega de proceso a proceso. Aunque la capa de red (mediante el protocolo IP) entrega datos a un dispositivo específico, es la capa de transporte la que decide a qué aplicación o servicio concreto pertenecen esos datos dentro de dicho dispositivo.

La multiplexación ocurre en el host de origen. Su función consiste en reunir fragmentos de datos provenientes de diferentes procesos de aplicación, añadirles un encabezado de transporte con identificadores específicos como el número de puerto y pasar los segmentos resultantes a la capa de red. Gracias a esto, múltiples aplicaciones como un navegador web, un cliente de correo y una videollamada pueden enviar datos simultáneamente a través de una única conexión física de red sin que la información se mezcle de forma irrecuperable.

La demultiplexación es el proceso inverso y ocurre en el host de destino. Cuando la capa de transporte recibe un segmento de datos de la capa de red, examina el número de puerto de destino en el encabezado. Luego, dirige esos datos al socket o punto de acceso correspondiente a la aplicación que los está esperando. De esta manera, el sistema operativo asegura que los datos del servidor web lleguen al navegador y no al programa de correo electrónico, por dar un ejemplo.

Existen diferencias en cómo se realiza este proceso dependiendo del protocolo de trasporte utilizado. En el caso de UDP, la demultiplexación se basa únicamente en la dirección IP de destino y el número de puerto de destino. Esto significa que si dos segmentos tienen el mismo puerto de destino pero vienen de diferentes orígenes, serán dirigidos al mismo socket. Por el contrario, en TCP la demultiplexación es más específica ya que utiliza una tupla de cuatro valores: dirección IP de origen, número de puerto de origen, dirección IP de destino y número de puerto de destino. Esto permite que un servidor mantenga múltiples conexiones simultáneas con diferentes clientes en el mismo puerto, como el puerto 80 o 443, manteniendo cada flujo de datos totalmente independiente del otro.

TCP

El protocolo de control de transmisión (Transmission Control Protocol, TCP) es un protocolo de transporte del modelo TCP/IP cuya función es permitir la comunicación confiable y ordenada entre aplicaciones que se ejecutan en distintos dispositivos de una red.

A diferencia de protocolos más simples como UDP, TCP es un protocolo orientado a la conexión. Esto significa que antes de que los datos comiencen a fluir, los dos dispositivos deben realizar un procedimiento de apertura llamado acuerdo de tres pasos o “three-way handshake”. Durante este proceso, ambos extremos sincronizan sus números de secuencia y confirman que están listos para intercambiar información. Cuando se desea terminar la comunicación, los dispositivos también deben realizar un procedimiento para terminar la conexión.

Una de las características más críticas de TCP es la fiabilidad mediante la detección de errores y la retransmisión. Cada segmento de datos que se envía lleva un número de secuencia único. Cuando el receptor recibe un paquete, devuelve un mensaje de confirmación llamado ACK. Si el emisor

no recibe esta confirmación en un tiempo determinado, asume que el paquete se perdió o se dañó y lo reenvía automáticamente. Esto asegura que la aplicación de destino reciba una copia exacta de los datos originales, sin importar las condiciones de la red. Además de la fiabilidad, TCP gestiona el control de flujo y el control de congestión. El control de flujo evita que un emisor rápido abrume a un receptor lento mediante el uso de una ventana deslizante, que indica cuánto espacio tiene disponible el receptor en su memoria temporal. El control de congestión, por su parte, permite que TCP reduzca la velocidad de envío si detecta que la red está saturada, evitando así un colapso del tráfico. Finalmente, TCP garantiza la entrega en orden. Debido a que Internet es una red de conmutación de paquetes, es común que diferentes fragmentos de un archivo sigan rutas distintas y lleguen al destino en desorden. TCP utiliza los números de secuencia mencionados previamente para reensamblar los segmentos en la posición correcta antes de entregarlos a la capa de aplicación, de modo que se reciba la información tal como fue enviada originalmente.

Inicio y cierre de sesión

Como se mencionó previamente, al iniciar una sesión TCP se sigue un acuerdo de 3 pasos ("three-way handshake"), esto comienza cuando un cliente desea establecer una conexión con un servidor, para esto envía un segmento TCP con la bandera (o campo, en el encabezado) SYN activada e incluye su numero de secuencia (el cual elige el sistema operativo mediante un algoritmo), si el servidor esta disponible responde con un segmento con las banderas SYN y ACK activas y también devuelve su numero de secuencia el cual es elegido de la misma forma (por el sistema operativo) además el servidor asigna a su campo ACK el numero de secuencia que el cliente envió en el primer segmento +1 ya que el primer segmento ocupa 1 byte. Por ultimo el cliente envía un segmento con la bandera ACK activa y con numero de ACK igual al numero de secuencia que el servidor envío en su respuesta +1, debido a que el segmento enviado por el servidor ocupa 1 byte.

El cierre de sesión TCP se da en 2 pasos, como se muestra en la figura 2 quien inicia el cierre de conexión envía un segmento TCP con el bit FIN activado, cuando recibe reconocimiento del otro ex-

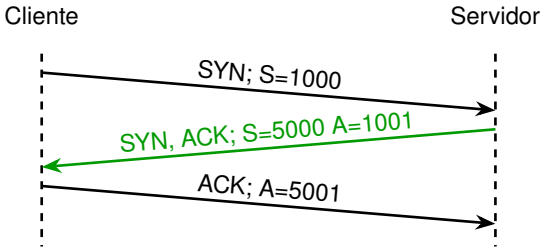

Fig. 1: Inicio de una conexión TCP

tremo de la conexión espera a recibir un segmento separado con el bit FIN activado, cuando llega este segmento envía un segmento de reconocimiento y se da por finalizada la sesión en ambos extremos.

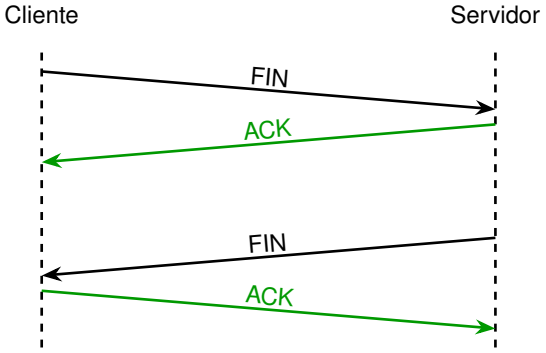

Fig. 2: Cierre de una conexión TCP

Control de flujo y congestion

El control de flujo controla el flujo de datos que se envía a la aplicación, ya que en aplicaciones lentas puede que el emisor desborde el buffer de recepción.

El control de congestión controla que no haya congestión en la red, esto puede provocarse debido a los protocolos subyacentes al nivel de transporte, como el protocolo IP.

En ambos casos se regula la velocidad de transmisión del emisor para mitigar los efectos y lograr una conexión segura.

Ventana deslizante

La ventana deslizante (sliding window) en TCP es el mecanismo de control de flujo que regula cuántos bytes puede enviar un emisor sin recibir confirmación (ACK) del receptor.

La ventana deslizante permite que el emisor envié multiples segmentos seguidos sin esperar el

segmento de validación de receptor instantáneamente.

Se implementa

Slow start

Fast retransmit

En el caso de que se reciban tres ACK con igual numero de reconocimiento, el emisor TCP realiza una retransmisión rápida lo cual permite retransmitir un segmento perdido sin esperar a que expire el temporizador de retransmisión (RTO). Un ejemplo de esto se muestra en la figura 3.

Cuando TCP detecta los ACK duplicados guarda esos segmentos en un buffer ya que detecta que los segmentos recibidos no están en orden y debe esperar a que el emisor retransmita el segmento perdido.

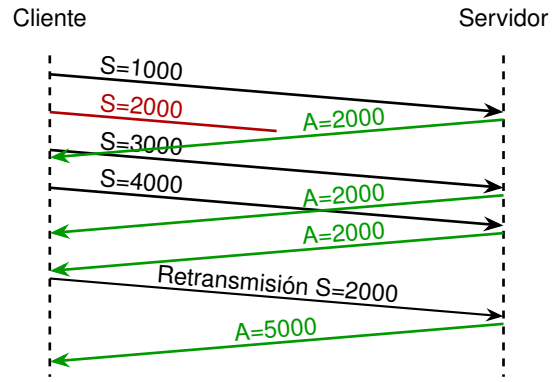

Fig. 3: Ejemplo de Fast Retransmit en TCP

Transmisión de datos

La flag `PSH` indica al receptor que debe enviar los datos recibidos inmediatamente a la aplicación, sin almacenarlos en un buffer. Esto se utiliza cuando en un segmento se envía la totalidad de la petición, por lo que no hace falta que el receptor espere a mas datos.

Maximum segment size

El tamaño máximo de segmento (Maximum Segment Size, MSS) es un parámetro del protocolo TCP que indica la cantidad máxima de bytes de datos útiles de la aplicación (payload) que puede transportar un segmento TCP, sin incluir cabeceras TCP ni IP.

Se define durante el three-way handshake al inicio de la conexión TCP y esta relacionado con el MTU del enlace. Por ejemplo, para un enlace Ethernet con MTU de 1500 bytes, y tomando una cabecera TCP e IPv4 de 20 bytes cada una se tiene un MSS de 1460 bytes.

$$MSS = 1500\ \text{B} - 20\ \text{B} - 20\ \text{B} = 1460\ \text{B}$$

Segmentación

Cuando la aplicación desea transmitir una cierta cantidad de bytes el protocolo TCP divide esos bytes en segmentos de acuerdo al tamaño máximo de segmento (MSS) del enlace por el cual se va a realizar la transmisión. Por ejemplo para transmitir 5000 bytes de datos con un MSS de 1460 bytes, se necesitaran 4 segmentos TCP, 3 segmentos de 1460 bytes y un segmento de 620 bytes. A cada segmento de datos se agrega una cabecera TCP, en este ejemplo sin opciones de 20 bytes, y una cabecera IP, también sin opciones en este ejemplo también ocupando 20 bytes, de esta forma se llega a los 1500 bytes de segmento máximo.

UDP

El Protocolo de datagramas de usuario (User Datagram Protocol, UDP) es un protocolo de transporte cuya función es enviar datagramas de forma rápida y simple, sin establecer conexión previa y sin garantizar entrega, orden ni control de errores.

QUIC

Nivel de red

La función principal de la capa de red es la de transportar paquetes desde un host emisor a un host receptor. Esta capa es muy amplia y se puede dividir conceptualmente en un **plano de control** y un **plano de datos**. El plano de datos (o forwarding plane) se encarga de reenviar un paquete que entra por una interfaz a otra interfaz apropiada, basándose en la tabla de forwarding o reenvío (o Forwarding Information Base, FIB) la cual ya ha sido previamente completada por el plano de control. El plano de control se encarga justamente de completar la tabla de reenvío para que así el plano de datos reenvíe los paquetes por la interfaz apropiada. Para completar la tabla de reenvío el enfoque tradicional, o más común, es elegir de acuerdo a la dirección de destino del paquete y basarse en la

tabla de ruteo (o Routing Information Base, RIB) para elegir la interfaz de salida. En otros enfoques no tradicionales, como enrutamiento basado en políticas (Policy Based Routeing, PBR) puede que se tenga en cuenta otros parámetros del paquete ademas de la dirección de destino, como la dirección de origen, y en ese caso la tabla de reenvío no dependería solamente de la tabla de ruteo.

En la tabla 2 se muestra un ejemplo de tabla de ruteo. En el enfoque tradicional, en el cual el reenvío se basa solo en la dirección de destino del paquete, se realiza la operación binaría AND entre la mascara de la entrada en la tabla y la dirección de destino del paquete entrante, luego se compara con la dirección de destino de la tabla y se reenvía de acuerdo a la interface de esa entrada en la tabla.

Una entrada particular de la tabla es lo que se conoce como **default gateway**, el cual típicamente es la entrada con todos ceros y mascara cero: 0.0.0.0/0, esta entrada es aquella por la cual debe salir el paquete si no coincide con ningún otra entrada en la tabla, de no estar esta entrada el paquete se descartaría ya que no existiría destino conocido por el cual reenviar el paquete.

Destino	Mascara	Próximo salto	Interface
123.0.0.0	/8	123.0.0.99	eth0
190.4.28.0	/16	190.4.28.1	eth0
200.10.20.0	/24	200.10.20.2	eth0
0.0.0.0	/0	200.10.20.1	eth0

Tabla 2: Ejemplo de tabla de ruteo

Puede ocurrir que una dirección IP coincida con una o más entradas de la tabla de ruteo, en cuyo caso, la entrada que tenga más campos en común con la dirección será la elegida. Por ejemplo, una dirección coincide con una entrada de la tabla con mascara /24 y también con una entrada con mascara /26, entonces el datagrama se reenvía por la interfaz correspondiente a la entrada de la tabla con la mascara /26.

Protocolo IP

El protocolo de internet (Internet Protocol, IP) es el componente principal de la capa de red y la principal función es la de actuar como un sistema de direccionamiento y enrutamiento universal, permitiendo que los datos viajen desde un origen hasta un destino a través de múltiples redes interconectadas. Se puede decir que el protocolo IP pertenece al plano de datos aunque depende de la información brindada por plano de control para direccionar los datagramas.

A diferencia de protocolos de capas superiores que se encargan de la fiabilidad, IP es un protocolo de entrega de mejor esfuerzo (*best-effort*) y no orientado a la conexión. Esto significa que no garantiza que los datos lleguen a su destino ni que lo hagan en el orden correcto; simplemente se encarga de empaquetar la información en unidades denominadas datagramas y enviarlas a través de la infraestructura de red basándose en las direcciones de los encabezados.

Cada datagrama IP contiene una sección de encabezado que incluye información vital para el transporte, siendo las direcciones IP de origen y de destino los campos más importantes. Estas direcciones identificando de forma única a cada dispositivo conectado a la red. Existen dos versiones del protocolo IP la version 4 y la version 6. La version 4 (o IPv4) utiliza direcciones de 32 bits representadas por cuatro bytes separados por punto, por ejemplo 192.168.1.1. La versión 6 (o IPv6) utiliza direcciones de 128 bits, con una representación en hexadecimal de ocho grupos de 16 bits separados por dos puntos, con reglas de compresión para los ceros, por ejemplo 2001:0db8:85a3:0000:0000:8a2e:0370:7334.

Versión 4

El formato de los datagramas de IPv4 se muestra en la tabla 3. Un datagrama IP tiene un total de 20 bytes de cabecera cuando no se utilicen opciones. Por lo general no se utilizan opciones en el encabezado IP ya que la mayoría de estas opciones son obsoletas. Si el datagrama transporta un segmento TCP, entonces cada datagrama (no fragmentado) transporta un total como mínimo de 40 bytes de cabecera (20 bytes de la cabecera IP más 20 bytes de la cabecera TCP, sin opciones en los dos casos) junto con el mensaje de la capa de aplicación.

Los campos de los datagramas IP version 4 son:

1. Número de versión: Ocupa 4 bits del encabezado y especifica la versión del protocolo IP del datagrama. A partir del número de ver-

0-3	4-7	8-15	16-32
Version	Header length	ToS	Total length
Identification		Flags	Fragment Offset
TTL		Protocol	Header checksum
Source address			
Destination address			
Options			
Padding			

Tabla 3: Formato encabezado IPv4

sión, el router puede determinar cómo interpretar el resto del datagrama IP.

2. Longitud de cabecera: Un datagrama IPv4 puede contener un número variable de opciones, estos 4 bits son necesarios para determinar dónde comienzan realmente los datos (o carga útil) en el datagrama IP. La mayoría de los datagramas IP no contienen opciones, por lo que el datagrama IP típico tiene una cabecera de 20 bytes.
3. Tipo de servicio: Los bits del tipo de servicio (Type Of Service, ToS) se incluyeron en la cabecera con el fin de poder diferenciar entre los distintos tipos de datagramas IP. Por ejemplo, para diferenciar datagramas en tiempo real (como los utilizados en aplicaciones de telefonía IP) del tráfico que no es en tiempo real (como por ejemplo el tráfico FTP).
4. Longitud del datagrama: Es la longitud total del datagrama IP (la cabecera más los datos) en bytes. Puesto que este campo tiene una longitud de 16 bits, el tamaño máximo teórico del datagrama IP es de 65535 bytes. Aunque rara vez tienen una longitud mayor a 1500 bytes.
5. Identificador, flags, offset: Estos campos se utilizan cuando hay fragmentación de IP.
6. Tiempo de vida (Time-To-Live, TTL): se incluye con el fin de garantizar que los datagramas no estarán eternamente en circulación a través de la red (debido, por ejemplo, a un bucle de enrutamiento de larga duración). Este campo se decrementa en una unidad cada vez que un router procesa un datagrama. Si el campo TTL alcanza el valor 0, el datagrama tiene que ser descartado por el router.
7. Protocolo: Este campo solo se utiliza cuando el datagrama IP alcanza su destino final. El valor de este campo indica el protocolo específico de la capa de transporte al que se pasarán los datos. Por ejemplo, un valor de 6 indica que los datos se pasan a TCP, mientras que un valor igual a 17 indica que los datos se pasan a UDP.
8. Suma de comprobación de cabecera: La suma de comprobación de cabecera ayuda a los routers a detectar errores de bit en un datagrama dado. Un router calcula la suma de comprobación de cabecera para cada datagrama IP recibido y detecta una condición de error si la suma de comprobación incluida en la cabecera del datagrama no coincide con la suma de comprobación calculada. Normalmente, los routers descartan los datagramas en los que se ha detectado que existe un error. La suma de comprobación tiene que volver a calcularse y almacenarse en cada router, ya que el campo TTL, y posiblemente también el campo de opciones, cambian.
9. Direcciones IP de origen y de destino: Cuando un origen crea un datagrama, inserta su dirección IP en el campo de dirección IP de origen e inserta la dirección del destino final en el campo de dirección IP de destino. A menudo, el host de origen determina la dirección de destino mediante una búsqueda DNS
10. Datos (carga útil): En la mayoría de las circunstancias, el campo de datos del datagrama IP contiene el segmento de la capa de transporte (TCP o UDP) que va a entregarse al destino. Sin embargo, el campo de datos puede transportar otros tipos de datos, como por ejemplo mensajes ICMP.

Fragmentación

La cantidad máxima de datos que una trama de la capa de enlace puede transportar se conoce como **unidad máxima de transmisión** (Maximum Transmission Unit, MTU). Dado que cada datagrama IP se encapsula dentro de una trama de la capa de enlace para ir de un router al siguiente, la MTU del protocolo de la capa de enlace impone un límite estricto a la longitud de un datagrama IP. Además en cada enlace a lo largo de la ruta entre el emisor y el receptor puede utilizar diferentes protocolos, los cuales imponen MTU diferentes.

Si el datagrama no cabe en una trama de la capa de enlace entonces se debe fragmentar la carga útil del datagrama IP en dos o más datagramas IP

más pequeños, esto es encapsular cada uno de los datagramas IP más pequeños en una trama de la capa de enlace distinta y enviar dichas tramas a través del enlace de salida. Cada uno de estos datagramas más pequeños se conocen como **fragmentos**. Cuando un host de destino recibe una serie de datagramas procedentes del mismo origen, tiene que determinar si algunos de esos datagramas son fragmentos de algún otro datagrama original más grande. Si algunos datagramas son fragmentos, tiene que determinar además cuándo ha recibido el último fragmento y cómo debe ensamblar los fragmentos que ha recibido para formar el datagrama original. Para que el host de destino pueda reensamblar los fragmentos, en la cabecera de IPv4 se incluye los campos identificación, indicadores y offset. Cuando se crea un datagrama, el host emisor marca el datagrama con un número de identificación, así como con las direcciones de origen y de destino. Normalmente, el host emisor incrementa el número de identificación para cada datagrama que envía. Cuando un router necesita fragmentar un datagrama, cada datagrama resultante (es decir, cada fragmento) se marca con la dirección de origen, la dirección de destino y el número de identificación del datagrama original. Cuando el destino recibe una serie de datagramas procedentes del mismo host emisor, puede examinar los números de identificación de los datagramas para determinar cuáles de ellos son fragmentos de un mismo datagrama más largo. Puesto que IP es un servicio no fiable, es posible que uno o más de los fragmentos nunca lleguen a su destino. Por esta razón, con el fin de que el host de destino esté absolutamente seguro de que ha recibido el último fragmento del datagrama original, ese último fragmento tiene un bit indicador de más fragmentos (More Fragments, MF) puesto a 0, mientras que los demás fragmentos tienen el bit MF puesto a 1. Además, para que el host de destino determine si falta un fragmento (y también para que pueda reensamblar los fragmentos en el orden apropiado), se utiliza el campo offset para especificar en qué posición dentro del datagrama IP original encaja el fragmento.

Direccionamiento

Las direcciones IPv4 tienen una longitud de 32 bits (lo que equivale a 4 bytes), por lo que existen un total de 232 (unos 4.000 millones) direcciones IP posibles. Estas direcciones normalmente se expresan utilizando la denominada notación decimal con puntos, en la que cada byte de la dirección se escribe en formato decimal y se separa mediante un punto del resto de los bytes de la dirección. Por ejemplo, la dirección IP 193.32.216.9. El 193 es el número decimal equivalente a los 8 primeros bits de la dirección; el 32 es el equivalente decimal de los segundos 8 bits de la dirección, y así sucesivamente. Por tanto, la dirección 193.32.216.9 en notación binaria se expresa como sigue:

```
11000001.00100000.11011000.00001001
```

Subnetting y Supernetting

Subnetting es el proceso de dividir una red IP en subredes más pequeñas mediante la extensión del prefijo del campo de red, para mejorar la eficiencia y administración del direccionamiento. Por cada bit tomado se divide la red en dos subredes con la mitad de capacidad de hosts. Por ejemplo se tiene la red 192.168.1.0/24 la cual tiene capacidad para 253 hosts (debido a que la dirección 0 representa la red y la dirección 255 se usa para broadcast). Tomando un bit más de mascara se tienen dos subredes 192.168.1.0/25 y 192.168.1.128/25 ambas con capacidad para 126 hosts.

Supernetting es el proceso inverso al Subnetting, es reducir el numero de bits que ocupa la mascara agrupando multiples redes. Por ejemplo se tienen las subredes 192.168.0.0/24 y 192.168.1.0/24 estas subredes ambas tienen capacidad para 253 hosts, haciendo supernetting tomando un bit menos de mascara, resulta en la red 192.168.0.0/23 lo cual tiene capacidad para 510 hosts.

Direccionamiento con y sin clases

En el direccionamiento **con clases** se establece una longitud fija para la parte de red de las direcciones IP (la mascara), esta longitud fija puede ser de 8, 16 y 24 bits correspondiente con la clase A, B y C, respectivamente (también existen las clases D y E pero no es común utilizarlas). Dada una dirección IP y sabiendo que se utiliza direccionamiento con clases se puede determinar la mascara de red observando el primer octeto de la dirección IP, como se muestra en la tabla 4.

Esta forma de particionar las direcciones, utilizando clases es ineficiente debido a que en mu-

Clase	Primer octeto	Máscara
A	0xxxxxxx (0 – 127)	/8
B	10xxxxxx (128 – 191)	/16
C	110xxxxx (192 – 223)	/24
D	1110xxxx (224 – 239)	No aplica
E	1111xxxx (240 – 255)	No aplica

Tabla 4: Clases IPv4

chos casos, como organizaciones pequeñas, una red con direccionamiento clase C es muy pequeña (254 hosts) y una red clase B es muy grande (65634 hosts). En el direccionamiento **sin clases** (Classless Interdomain Routing, CIDR) se utilizan prefijos de longitud variable, eliminando las clases y mejorando la eficiencia, ya que se pueden tener subredes de tamaño arbitrario, este ultimo método es el que se usa en la actualidad, la única desventaja con el método con clases es que se debe especificar siempre con que longitud de mascara se esta trabajando.

Al igual que sucede con el direccionamiento de subredes, la dirección IPv4 de 32 bits se divide en dos partes y de nuevo se expresa en notación decimal con puntos como a.b.c.d/x, donde x indica el número de bits de la primera parte de la dirección. Los x bits más significativos de una dirección en el formato a.b.c.d/x constituyen la parte de red de la dirección IP y a menudo se los denomina prefijo (o prefijo de red) de la dirección.

La dirección IPv4 255.255.255.255 (en binario todos los campos en 1) se denomina dirección de difusión (o broadcast) y cuando un host envía un datagrama cuya dirección de destino es 255.255.255.255, el mensaje se entrega a todos los hosts existentes en la misma subred.

Por ejemplo se tiene la siguiente red de la cual se tiene 255 direcciones de host disponibles y una única red, pero se desea particionar la red en 4 subredes.

```
192.168.1.0/24
```

Para esto se hace una subred tomando los primero 26 bits de la dirección, resultando en una mascara /26 (en lugar de los primeros 24 bits de dirección para la clase, como indica la masca /24). El resultado son 4 subredes con espacio para 62 host cada subred.

```
192.168.1.0/26
192.168.1.64/26
192.168.1.128/26
192.168.1.192/26
```

DHCP

Las direcciones de host también se pueden configurar manualmente, pero frecuentemente esta tarea se lleva cabo utilizando el Protocolo de configuración dinámica de host (DHCP, Dynamic Host Configuration Protocol).DHCP permite a un host obtener (permite que se le asigne) automáticamente una dirección IP. Un administrador de red puede configurar DHCP de modo que un host dado reciba la misma dirección IP cada vez que se conecte a la red, o bien a un host puede asignársele una dirección IP temporal que será diferente cada vez que el host se conecte a la red. Además de la asignación de direcciones IP de host, DHCP también permite que un host obtenga información adicional, como por ejemplo su máscara de subred, la dirección de su router del primer salto [a menudo denominado pasarela (gateway) predeterminada] y la dirección de su servidor DNS local

Versión 6

En la tabla 5 se muestra el formato de un datagrama IP de versión 6. Esta versión aumenta el tamaño de la dirección IP, de 32 a 128 bits. Además de las direcciones de unidifusión y de multidifusión, IPv6 ha introducido un nuevo tipo de dirección, denominado dirección **anycast**, que permite que múltiples nodos tengan la misma dirección IPv6 y que el tráfico llegue automáticamente al nodo más cercano o más óptimo. También se eliminan algunos de los campos de la version anterior, o se han hecho opcionales, de cualquier forma la cabecera IPv6 es de longitud fija de 40 bytes a diferencia de la version anterior.

A diferencia de la version anterior no se permite ni la fragmentación ni el reensamblado en routers intermedios; estas operaciones solo pueden ser realizadas por el origen y el destino. Si un router recibe un datagrama IPv6 y es demasiado largo para ser reenviado por el enlace de salida, el router simplemente lo descarta y envía de vuelta al emisor un mensaje de error ICMP “Paquete demasiado grande”. El emisor puede entonces reenviar los datos utilizando un tamaño de datagrama IP más pequeño..

Version	Traffic class	Flow label	
Payload length		Next header	Hop limit
Source address			
Destination address			

Tabla 5: Formato encabezado IPv6

La cabecera IPv6 dispone de los siguientes campos:

1. Versión (4 bits): Identifica el número de versión del protocolo IP utilizado.
2. Clase de tráfico (8 bits): Al igual que el campo ToS de IPv4, el campo de clase de tráfico, se utiliza para dar prioridad a ciertos datagramas dentro de un flujo, o para dar prioridad a los datagramas de determinadas aplicaciones (por ejemplo, VoIP) frente a los datagramas de otras aplicaciones (como SMTP).
3. Etiqueta de flujo (20 bits): Este campo se utiliza para identificar un flujo de datagramas.
4. Longitud de la carga útil (16 bits): Este valor se trata como un entero sin signo que proporciona el número de bytes del datagrama IPv6 incluidos a continuación de la cabecera del datagrama.
5. Siguiente cabecera (8 bits): Este campo identifica el protocolo (por ejemplo, TCP o UDP) al que se entregará el contenido (el campo de datos) de este datagrama. El campo utiliza los mismos valores que el campo de protocolo de la cabecera IPv4.
6. Límite de saltos (8 bits): Cada router que reenvía un datagrama decrementa el contenido de este campo en una unidad. Si el límite de saltos alcanza el valor cero, el datagrama se descarta.
7. Direcciones de origen y de destino (128 bits cada una): se usan para identificar quién envía el paquete y a quién debe llegar

Protocolo ICMP

El Protocolo de Mensajes de Control de Internet (Internet Control Message Protocol, ICMP) es un protocolo de control y diagnóstico de la capa de red que se utiliza para informar errores y condiciones de funcionamiento de la red. Los mensajes ICMP si bien corresponden a la capa de red se encapsulan dentro de datagramas IP. Existen dos versiones de ICMP, uno para IPv4, ICMP y otro para IPv6, ICMPv6.

Los hosts y los routers utilizan ICMP para intercambiarse información acerca de la capa de red. El uso más típico de ICMP es la generación de informes de error. Por ejemplo, al ejecutar una sesión HTTP podemos encontrarnos con un mensaje de error como “Red de destino inalcanzable”. Este mensaje tiene su origen en ICMP. En algún punto, un router IP no ha podido encontrar una ruta hasta el host especificado en la solicitud HTTP, y dicho router ha creado y enviado un mensaje ICMP a nuestro host para informarle del error.

Los mensajes ICMP tienen un campo de tipo y un campo de código, y contienen la cabecera y los 8 primeros bytes del datagrama IP que ha dado lugar a la generación del mensaje ICMP (para que el emisor pueda determinar qué datagrama ha producido el error)

NAT

El protocolo de traducción de direcciones de red (Network Address Translation, NAT) es un protocolo que permite que varios dispositivos en una red privada compartan una única dirección IP pública.

Una dirección **IP pública** es única y accesible mundialmente, mientras que una dirección **IP privada** se usa solo dentro de una red local y no es accesible fuera de esa red. Las redes publicas son accesibles por los proveedores de internet, mientras que las redes privadas son asignadas mediante los administradores de la red privada o mediante servidores DHCP, por ejemplo.

Ruteo

El ruteo es el proceso de calculo y selección de rutas optimas hacia los destinos. El proceso de ruteo ocurre en el plano de control. El ruteo puede ser estático, dinámico o predeterminado. En el ruteo estático las rutas se configuran manualmente y no cambian a menos que un administrador de red lo haga. En el ruteo dinámico las rutas se ajustan automáticamente según el estado de la red, para

esto se utilizan protocolos de ruteo que modifican las rutas basándose por ejemplo en el camino más corto. El ruteo predeterminado se usa cuando no hay una ruta especifica hacia un destino y se envía por ese camino por defecto.

Un **sistema autónomo** (Autonomous System, AS) es un conjunto de redes IP que están bajo un mismo control administrativo y que comparten una política de ruteo común.

El ruteo se puede dividir en **ruteo interno** y **ruteo externo** de acuerdo al alcance y tipos de protocolos que se utilizan para establecer las rutas. El ruteo interno (Interior Gateway Protocol, IGP) se realiza dentro de un mismo sistema autónomo, por ejemplo dentro de una empresa u organización. El ruteo externo (Exterior Gateway Protocol, EGP) se usa para intercambiar rutas entre diferentes sistemas autónomos, es decir, entre redes diferentes, generalmente a través de internet.

Los protocolos de ruteo definen como se deben intercambiar los mensajes con los nodos adyacentes, el formato del mensaje, cada cuanto se intercambian los mensajes, etc. con la información recolectada sobre la red el protocolo ejecuta un algoritmo, denominado **algoritmo de ruteo**, el cual decide cual será la ruta más optima para cada destino.

Algoritmos de ruteo

Los algoritmos de ruteo tienen como objetivo determinar buenas rutas desde los emisores a los receptores, a través de la red de routers de manera tal de calcular la tabla de enrutamiento más optima. Normalmente, una “buena ruta” es aquella que tiene el coste mínimo. El **coste mínimo** es la ruta cuyo valor total de métrica acumulada es el más bajo entre todas las posibles rutas hacia un destino, generalmente se toma la sumatoria del costo de todos los enlaces de la ruta. El costo de un enlace se asigna de acuerdo al protocolo de ruteo utilizado.

Protocolos de ruteo

Los protocolos de ruteo, como se mencionó previamente, definen como se intercambian los mensajes entre los nodos, cada cuanto tiempo, el formato del mensaje, etc., recolectan información de la red y ejecutan algoritmos sobre la información recolectada para determinar la ruta más optima. Los protocolos de ruteo se diferencian principalmente en internos y externos, los protocolos de ruteo interno (Interior Gateway Protocol, IRP) se utilizan dentro de un sistema autónomo mientras que los protocolos de ruteo externo (Exterior Gateway Protocol, EGP) se utilizan entre sistemas autónomos. Los protocolos de ruteo interno se pueden clasificar según el tipo de algoritmo que ejecutan, siendo los más conocidos los protocolos de distancia vectorial y los protocolos de estado de enlace.

Los protocolos de tipo vector distancia son más antiguos, o primitivos, el más conocido de este tipo es RIPv1 y la version actualizada RIPv2 (también existe para direcciones IPv6). Los algoritmos link state son más modernos y utilizan algoritmos más avanzados, por ejemplo el algoritmo de Dijkstra, para determinar la ruta más corta. El más protocolo más conocido de tipo link state es SPF y su version abierta OSPF.

Vector distancia

En los algoritmos de distancia vectorial la mejor ruta se determina aplicando el algoritmo de Bellman-Ford. Este algoritmo calcula los caminos mínimos desde un nodo origen hacia todos los demás nodos, para esto cada router debe mantener una tabla con destino, costo y next-hop. La tabla de cada nodo se actualiza de acuerdo a actualizaciones periódicas que recibe de sus nodos adyacentes.

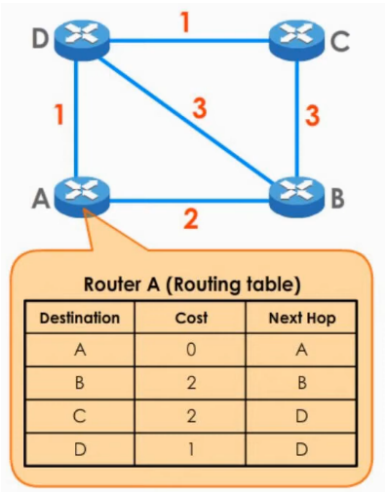

Fig. 4: Tabla de ruteo de un nodo

El mejor camino desde un punto a otro se determina de manera iterativa por la ecuación de

Bellman-Ford, la cual se muestra en (1) y dice que la mejor ruta desde *x* hasta *y* es aquella que pasa por el vecino *v* que minimiza la suma, es decir, la mejor ruta es aquella que minimiza el costo hacia el vecino sumado el menor costo del vecino hacia el destino. Dado que la ecuación de Bellman-Ford es recursiva, ya que depende del costo del nodo adyacente y este de su adyacente, etc. es posible que se formen lazos y la ecuación resulte infinita.

$$D_x(y) = \min_{v \in vecinos} \{ c(x,v) + D_v(y) \} \quad (1)$$

La métrica (o costo del enlace, también llamado distancia) se mide teniendo en cuenta parámetros como la velocidad del enlace, la latencia o condiciones de la red. En muchos casos simplemente se toma la métrica como la cantidad de saltos (o distancia) de un punto a otro, por ejemplo para un enlace punto a punto la métrica es 1.

RIP

El protocolo de información de rutas (Routing Information Protocol, RIP) fue uno de los primeros protocolos de ruteo interno dinámico basado en vector distancia. Utiliza como métrica la cantidad de saltos y se limita a 15 para prevenir el problema de lazos infinitos, esto pone un limita el tamaño de la red. Si la distancia entre un punto y otro en la red es 16, entonces se considera distancia **infinita** y por tanto ese punto es inalcanzable. El protocolo RIP implementa también los mecanismos de horizonte dividido (Split Horizon) y envenenamiento de ruta (Route Poisoning) para prevenir lazos.

Por ser un algoritmo de distancia vectorial los nodos compartes su tabla de ruteo con los nodos adyacentes mediante actualizaciones periódicas. En RIP las actualizaciones por defecto ocurren cada 30 segundos, aunque este es un parámetro que en muchos casos es configurable.

La version 1 del protocolo RIPv1 se publicó en 1988. Un router luego de configurarse el protocolo y pasados 30 segundos (el timer por defecto) emite un datagrama de petición de RIPv1 a la dirección broadcast (dirección 255.255.255.255) a todas las interfaces con RIPv1 habilitado. Luego los dispositivos corriendo el mismo protocolo responde con su tabla de ruteo.

Split horizon

Compensa los efectos de la cuenta a infinito Los routers no anuncian rutas por las interfaces que las aprendieron(recibieron)

En split horizon con poison reverse las rutas se publican pero con métrica infinito.

Link state

En los algoritmos de estado de enlace cada nodo dispone de un mapa completo de la red, y entre nodos adyacentes se intercambian sus mapa completos.

SPF/OSPF

El camino más corto primero (Shortest Path First, SPF) es un protocolo de ruteo interno de tipo link state que calcula el coste mínimo desde un router hacia todos los demás nodos en la red. En la practica es el algoritmo de Dijkstra aplicado a enrutamiento.

Policy-Based Routing

Los algoritmos de ruteo basados en políticas PBR permiten seleccionar la ruta de un paquete según criterios como origen, protocolo, puerto, tipo de tráfico o marca, en lugar de usar solo la tabla de ruteo tradicional.

SDN

En las Redes definidas por software (Software-defined Networks, SDN) los dispositivos de red (switches, routers, etc) no tienen funciones asignadas como en las redes tradicionales, si no que un controlador centralizado les asigna una función.

OpenFlow

El protocolo OpenFlow opera entre un controlador SDN y un conmutador controlado por SDN u otro dispositivo que implemente la API OpenFlow. El protocolo OpenFlow funciona sobre TCP, siendo su número de puerto predeterminado el 6653.

Nivel de enlace

Para que un datagrama pueda ser transferido desde el host de origen al de destino, debe moverse a través de cada uno de los enlaces individuales que forman la ruta extremo a extremo. En un determinado enlace, un nodo transmisor encapsula el datagrama en una trama de la capa de enlace y transmite la trama a través del enlace. En su

mayor parte, la capa de enlace se implementa en un adaptador de red, también denominado a veces tarjeta de interfaz de red (Network Interface Card, NIC). A los adaptadores de red se les asigna una dirección de la capa de enlace, un host o un router con múltiples interfaces de red tendrá asociadas, por tanto, múltiples direcciones de la capa de enlace.

A las direcciones de la capa de enlace se las denomina de multiples formas, como **dirección LAN**, **dirección física** o **dirección MAC**, siendo esta ultima la mas utilizada. En la mayoría de las redes LAN la dirección MAC tiene 6 bytes de longitud y se suelen expresar en notación hexadecimal, indicándose cada byte mediante una pareja de números hexadecimales y cada pareja separada por dos puntos o guiones, por ejemplo `00:1A:2B:3C:4D:5E` o `1A-23-F9-CD-06-9B`, pero existen otros formatos validos.

Protocolo ARP

Dado que existen tanto direcciones de la capa de red (por ejemplo, direcciones IP de Internet) como direcciones de la capa de enlace (es decir, direcciones MAC), surge la necesidad de una traducción entre ellas. En Internet, esta tarea la lleva a cabo el protocolo ARP (Address Resolution Protocol, Protocolo de resolución de direcciones)

Un módulo ARP en el host emisor toma como entrada cualquier dirección IP de la misma LAN y devuelve la dirección MAC correspondiente. En nuestro ejemplo, el host emisor 222.222.222.220 proporciona a su módulo ARP la dirección IP 222.222.222.222 y el módulo ARP devuelve la correspondiente dirección MAC 49:BD:D2:C7:56:2A.

Un mensaje ARP de consulta se envía dentro de una trama de difusión, mientras que el mensaje ARP de respuesta se envía dentro de una trama estándar.

Un paquete ARP se encapsula dentro de una trama de la capa de enlace y así se sitúa, arquitectónicamente, encima de la capa de enlace. Sin embargo, un paquete ARP dispone de campos que contienen direcciones de la capa de enlace, por lo que se podría decir que es un protocolo de la capa de enlace, pero también contiene direcciones de la capa de red y, por tanto, podría también argumentarse que es un protocolo de la capa de red.

Comprobación de errores

Quizá la forma más simple de detección de errores sea el uso de un único bit de paridad.

Redes LAN

Las redes de area local (Local Area Network, LAN) son redes de computadoras que históricamente cubren un área geográfica pequeña, como una habitación, una casa, un edificio o un campus reducido y utiliza tecnologías de alta velocidad como Ethernet o Wi-Fi.

Para acceder al medio en una red alámbrica existe un mecanismo denominado mecanismo de acceso múltiple por detección de portadora con detección de colisiones (Carrier Sense Multiple Access with Collision Detection, CSMA/CD)

Una **colisión** puede ocurrir al comienzo de la transmisión, cuando dos estaciones desean transmitir al mismo tiempo, censan el medio y detectan que el canal no esta ocupado, comienza a transmitir y se produce la colisión. Es por esto que por convención se considera que solo puedo ocurrir una colisión de este estilo durante la transmisión de los primeros 512 bits de la trama, ya que si no ocurrió una colisión en esos bits es correcto asegurar que no habrá colisión al transmitir los siguientes bits. Tomar por convención los primeros 512 bits de la trama fija una distancia maxima de enlace, la cual esta determinada por la velocidad de propagación de los bits por el enlace. Un host debe estar censando el medio por el tiempo que tarda la propagación de los 512 bits hasta el otro extremo de la red y vuelta.

Una retransmisión (algoritmo de back-off) solo ocurre si se detecta una colisión. Si un receptor detecta algún error en la trama **no** se retransmite, si no que se descarta esa trama.

Ethernet 802.3

Para transmitir en el protocolo Ethernet 802.3 half-duplex se siguen los siguientes pasos:

1. Se censa la red, si la red no esta disponible se espera a que lo esté.
2. Luego de detectar que la red esta disponible se espera un Interframe Gap (IFG) el cual por convención es una transmisión de 96 bits, este tiempo esta determinado por la velocidad de la

red, por ejemplo para 100 Mbps se tiene 0.96 microsegundos $\frac{96\ bits}{100\ bits/s}$
3. Se deshabilita la detección de colisiones.
4. Se transmite el preámbulo y delimitador de inicio de trama (start of frame delimiter, SFD). El preámbulo son 7 bytes de la forma 10101010 repetidos y el SFD es 10101011.
5. Se habilita la detección de colisiones.
6. Se transmiten los primeros 512 bits de la trama y durante este tiempo se detectan colisiones. Si ocurre una colisión se detiene la transmisión y se transmite una señal de Jam (señal de 32 bits) lo que hace que todas las estaciones en la red detecten la colisión y descarten la trama. Si no se detecta colisión durante estos primeros 512 bits, se deshabilita la detección de colisiones y se transmite el resto de la trama.

Si la red es full-duplex, el comportamiento cambia de forma fundamental respecto a Ethernet half-duplex.

1. No existe CSMA/CD: En full-duplex: No hay detección de portadora previa obligatoria. No existen colisiones. No se usa señal de jam. No hay backoff exponencial. CSMA/CD solo aplica a medios compartidos (bus o hubs).

2. Por qué no hay colisiones: En full-duplex el enlace es: Punto a punto (host ↔ switch). Con canales físicos separados: Un par para transmisión (TX). Un par para recepción (RX). Ambos dispositivos pueden transmitir simultáneamente sin interferencia eléctrica. No hay dominio de colisión.

En los switches modernos todas las interfaces trabajan en modo full-duplex.

VLAN

Las redes de area local virtuales (Virtual Local Area Network, VLAN) permiten usar la infraestructura existente para dar servicio a multiples redes (o dominios de broadcast). Se pueden implementan en uno o mas switches aunque existen routers que también permiten implementarlas. Estas redes virtuales permiten crear redes aisladas utilizando la misma infraestructura e incluso los mismos enlaces.

Las redes VLAN se identifican entre switches (dispositivos de la capa de enlace) para esto utilizan un formato especifico de trama Ethernet. Los puertos entre switches pueden ser de tipo troncal (trunk) o de acceso (access). A través de los enlaces de tipo troncal los switches se comunican utilizando tramas Ethernet con un campo especifico de VLAN, mientras que a través de los enlaces de acceso se comunican utilizando tramas Ethernet convencionales (sin un campo para VLAN).

Spanning Tree Protocol

Spanning Tree Protocol (STP) y la version mas moderna Rapid Spanning Tree Protocol (RSTP) sirven para evitar enlaces redundantes en redes Ethernet 802.3. Si se detecta un ciclo extra entre switches se ejecuta un algoritmo que deshabilita ese enlace redundante. Los bridges o switches que ejecutan el algoritmo intercambian información con mensajes de control denominados Bridge Protocol Data Unit (o BPDU). El BPDU siempre tiene dirección de destino :1:80:C2:00:00:00.

★ Detectan posibles ciclos entre switches.
★ Construyen una topología lógica sin bucles (en forma de árbol).
★ Bloquean enlaces redundantes.
★ Reactivan enlaces alternativos si falla el principal.

La diferencia principal entre ambas versiones es que la version RSTP converge mucho mas rápido a la topología sin bucles que STP, ya que es mas eficiente.

El bridge identifier (BID) es un identificador de cada bridge (o switch) que se determina por la dirección MAC del bridge y un numero de prioridad (de 2 bytes) configurable.

El root bridge (o switch) es el switch con el menor BID (aquel con el valor mas bajo) y puede haber solo un root bridge por dominio broadcast.

El root path cost es el camino con menor costo desde un switch, o segmento de red, hasta el root bridge. Se calcula sumando el costo de cada enlace. Todos los switches y enlaces de red tienen un root path cost asociado, y por convención el del root bridge es cero.

Estados de los puertos

El algoritmo RSTP asigna a cada puerto uno de tres estados:

★ Discarding (D): no participa en la topología de la red y no aprende direcciones MAC.
★ Learning (L): se aprenden las direcciones MAC pero no envía ni recibe tramas.
★ Forwarding (F): aprende direcciones MAC y envía y recibe tramas.

Roles de los puertos

El algoritmo RSTP asigna a cada puerto un rol:

★ Root port (RP): es el puerto donde se enviarán y recibirán tramas en dirección al root bridge.
★ Designated port (DP): es el puerto que envía el mejor BPDU al segmento donde esta conectado.
★ Alternate port (AP): provee un camino alternativo al provisto por el RP en dirección al root bridge.
★ Backup port (BP): provee un camino alternativo al provisto por el DP en dirección al root bridge.
★ Disabled port (DP): no se utiliza en la operación de RSTP. No recibe ni transmite BPDUs.

Redes inalámbricas

En redes inalámbricas los estándares que predominan son los del Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers, IEEE). En particular el estándar 802.11 de 1997 y todas sus modificaciones, como 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, etc. Por lo general los medios de transmisión inalámbricos son de naturaleza half-duplex.

Puntos de acceso

Un punto de acceso (Access Point, AP, o también Wireless Access Point, WAP) es un dispositivo de red que actúa como puente entre dispositivos inalámbricos (como celulares, laptops, etc.) y una red cableada, por lo general una red Ethernet.

Para que un dispositivo inalámbrico pueda transmitir datos en una red debe haber sido asociado y autenticado en la red por el AP.

En primer lugar el dispositivo sin autenticar y sin estar asociado comienza a mandar tramas de gestión (o probe requests) los cuales se envían en todos los canales y en todas las bandas un canal y una banda a la vez, de esta forma se descubren las redes inalámbricas.

Sistema distribuido

En el estándar 802.11, el sistema distribuido (o Distributed System, DS) es el sistema que interconecta múltiples puntos de acceso (APs) y permite que las distintas redes inalámbricas funcionen como una sola red.

MAC 802.11

El estándar MAC 802.11 impone un mecanismo de acceso al medio que permite el acceso justo y equitativo a todas las estaciones que lo deseen. En este estándar se utiliza acceso múltiple por detección de portadora con evitación de colisiones (Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA) en lugar de CSMA/CD para acceder al medio, esto porque los puntos de acceso (Access Point, AP) no tienen la posibilidad de detectar colisiones como si ocurre en los medios por cable. Puesto que no se pueden detectar las colisiones, el mecanismo CSMA/CA implementa un sistema en el que cada AP debe transmitir una trama de reconocimiento (Acknowledge, ACK) cuando recibe una trama.

Los principales mecanismos que hacen al estándar 802.11 son: el censado de portadora (Carrier Sense, CS), la función de coordinación distribuida (Distributed Coordination Function, DCF), las tramas de reconocimiento (Acknowledgment Frames, ACK) y las peticiones de envió y recepción (Request to Send/Clear to Send, RTS/CTS).

Carrier Sense

DCF

ACK

RTS/CTS

Según el estándar no es obligatorio implementarlo en el AP, pero en ciertos casos, como en el problema del nodo oculto, resulta útil para evitar colisiones.

Fragmentación

La fragmentación también es posible en el estándar 802.11 pero al igual que ocurría con RTS/CTS no es obligatorio su implementación.

La fragmentación tiene la desventaja que cada fragmento tiene su cabecera propia, lo que implica

una carga adicional de bytes. También entre fragmentos se debe esperar un SIFS y un ACK, por lo que también se agrega tiempo muerto que no se puede usar para transmitir datos útiles.