



Capa de Transporte



Capa de transporte

Los protocolos utilizados en la capa de transporte:

TCP (*Transmission Control Protocol*): protocolo de capa de transporte orientado a la conexión (PDU=segmentos)

UDP (*User Datagram Protocol*): protocolo de capa de transporte no orientado a la conexión (PDU=datagramas)



TCP (Transmission Control Protocol)

TCP es un protocolo de comunicación orientado a conexión, fiable a nivel de capa de transporte. Se diseñó específicamente para proporcionar un flujo de bytes confiable de extremo a extremo a través de una interred no confiable.

Es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 y 1974 por Vint Cerf y Robert Kahn.

Muchas aplicaciones pueden usar TCP para crear conexiones y enviar un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y reensamblados en el orden correcto.

Una entidad TCP acepta flujos de datos de usuario de procesos locales, los divide en fragmentos (agrega encabezado) que no excedan los 64 KB, y envía cada fragmento como un paquete IP independiente. La entidad TCP emisora y la receptora intercambian datos en forma de segmentos.



TCP (Transmission Control Protocol)

TCP da soporte a muchos protocolos populares de capa de aplicación, como HTTP, SMTP, SSH y FTP.

Proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

El servicio TCP se obtiene al hacer que el servidor y el cliente creen puntos terminales, llamados *sockets*. Cada socket tiene una dirección IP y un puerto, el cual es un número entero de 16 bits. Para obtener el servicio TCP, se debe establecer de manera explícita una conexión entre un *socket* en la máquina emisora y uno en la máquina receptora.

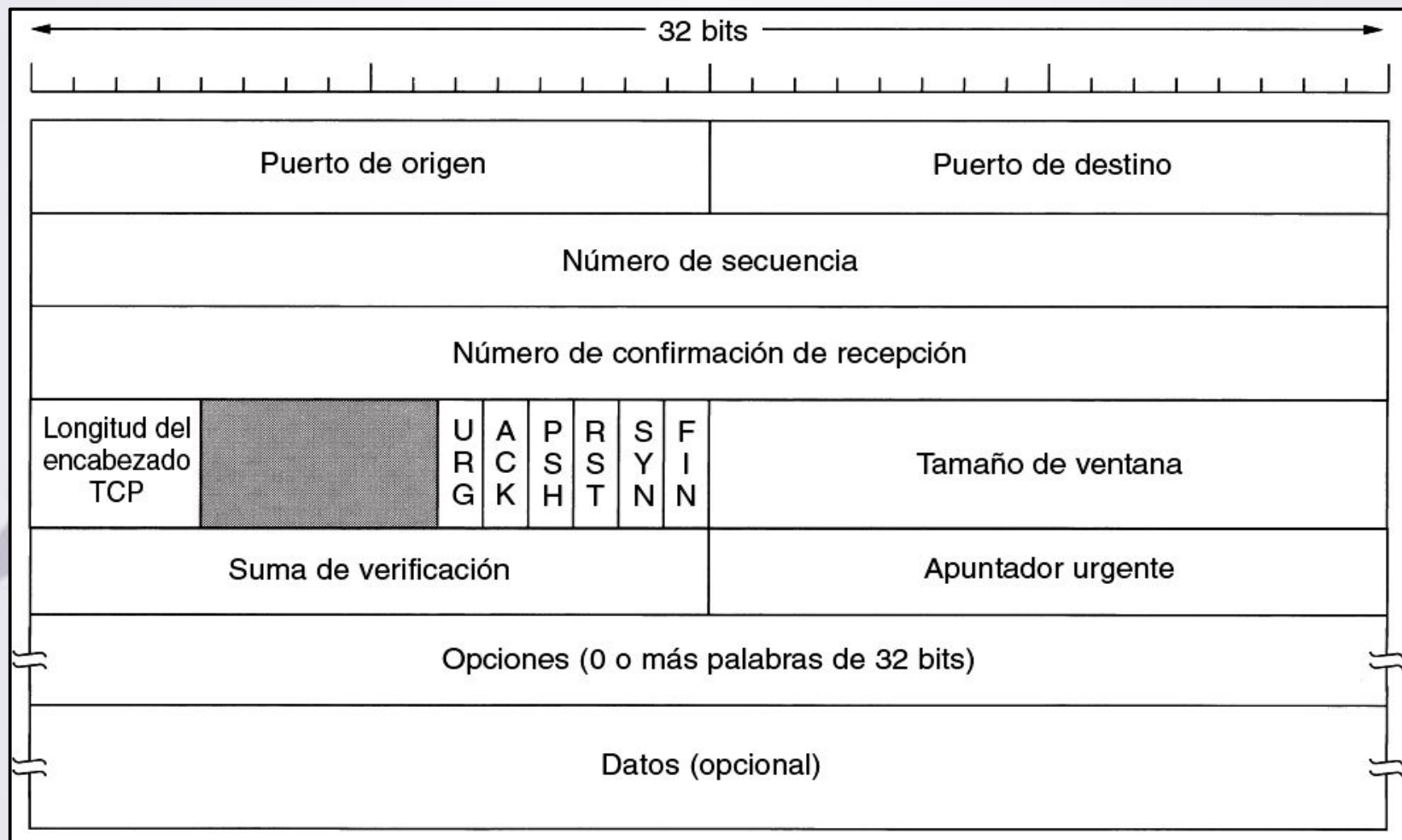
Todas las conexiones TCP son full dúplex y punto a punto.

TCP no soporta multidifusión ni difusión.



Protocolo TCP

Segmento TCP



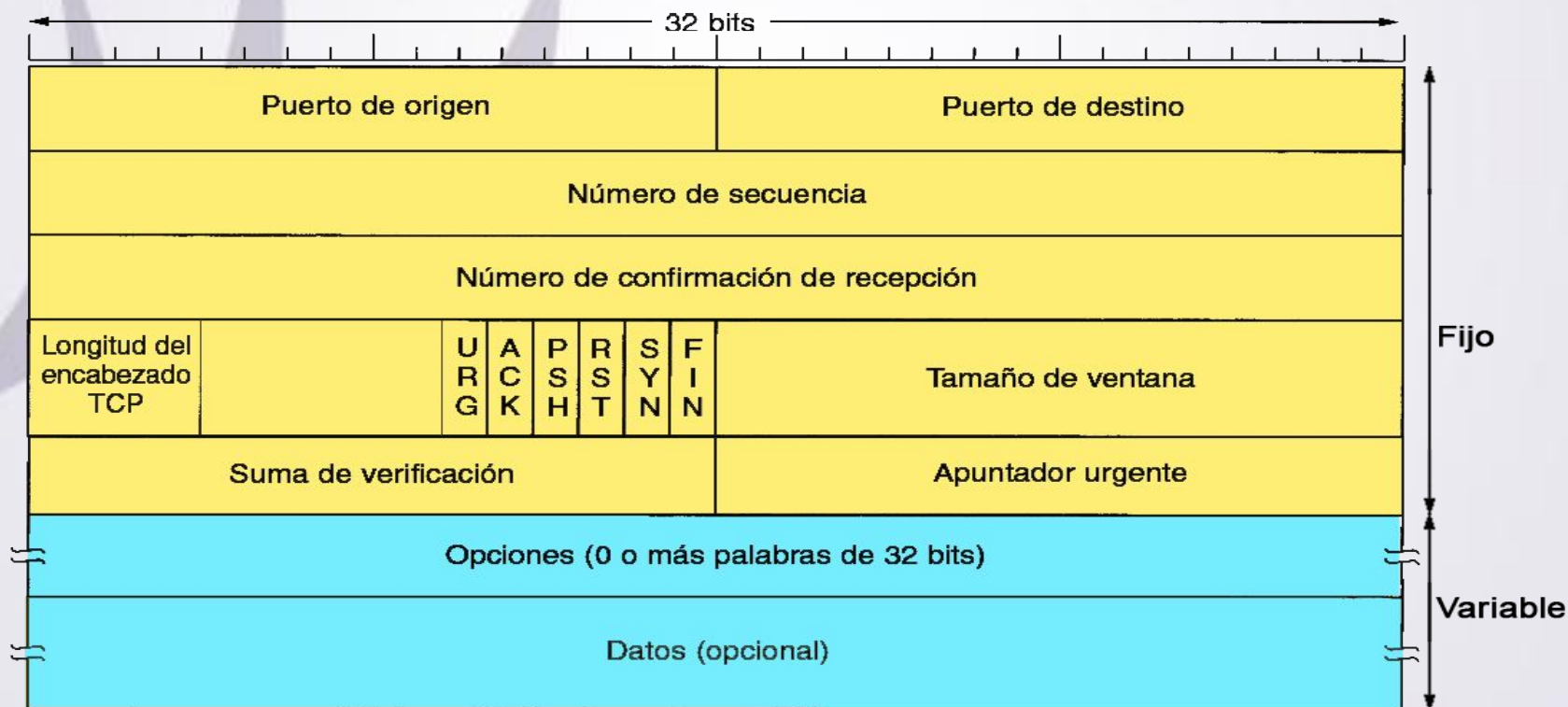


Protocolo TCP

Segmento TCP

Un segmento consiste en un encabezado TCP que tiene una parte fija de 20 bytes, más una parte opcional, seguido de cero o más bytes de datos.

Luego del encabezado fijo vienen las opciones. Si no las hay, pueden continuar hasta 65.495 bytes de datos como máximo ($65.535 - 20 - 20$), ya que los primeros 20 bytes se deben al encabezado IP y los segundos 20 bytes al encabezado TCP.





Segmento TCP

El campo **longitud de encabezado TCP** (4 bits) indica la cantidad de palabras de 32 bits contenidas en el encabezado

Banderas de 1 bit:

URG: campo Apuntador urgente es válido.

ACK: bit de acuse de recibo.

PSH: indica que los datos se deben transmitir de inmediato. Se solicita al receptor que entregue los datos a la aplicación a su llegada y no los almacene en buffer hasta la recepción de un buffer completo.

RST: restablecer una conexión, rechazar un segmento no válido o intentar abrir una conexión.

SYN: se usa para establecer conexiones.

FIN: el emisor ha llegado al final de su flujo de datos.



Segmento TCP

Los campos **Puerto de origen** y **Puerto de destino** identifican los puntos terminales locales de la conexión.

El campo **número de secuencia** identifica la posición de los datos del segmento en el flujo de datos del transmisor.

El campo **número de confirmación de recepción** identifica el valor del siguiente número de secuencia que el emisor del segmento espera recibir. (especifica el siguiente número de secuencia esperado, no el último número de secuencia correctamente recibido).



Segmento TCP

El control de flujo en el TCP se implementa usando una ventana corrediza de tamaño variable. El campo **tamaño de ventana** indica la cantidad de bytes que pueden enviarse, comenzando por el byte cuya recepción se ha confirmado.

Un tamaño de ventana de 0, indica que se han recibido los bytes hasta el número de confirmación de recepción -1 inclusive, pero que el receptor no puede recibir más datos por el momento.

El permiso para enviar puede otorgarse después enviando un segmento con el mismo número de confirmación de recepción y un campo tamaño de ventana distinto de cero.



Segmento TCP

Suma de verificación: Checksum utilizado para la comprobación de errores tanto en la cabecera como en los datos.

Puntero urgente: Cantidad de bytes desde el número de secuencia que indica el lugar donde acaban los datos urgentes.

Opciones: Para poder añadir características no cubiertas por la cabecera fija.

Relleno: Se utiliza para asegurarse que la cabecera tenga un tamaño múltiplo de 32 bits.



Establecimiento de una conexión TCP

En TCP las conexiones se establecen usando el acuerdo de tres vías.

Para establecer una conexión el servidor abre un socket en un determinado puerto TCP y espera pasivamente una conexión.

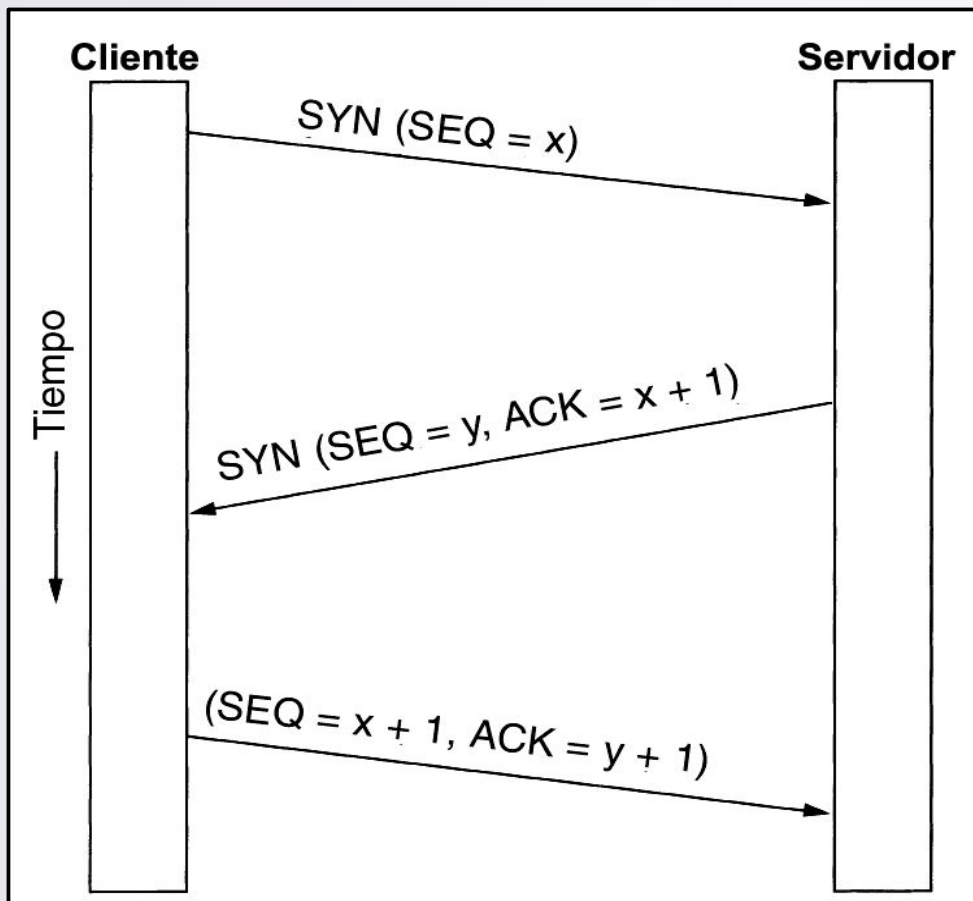
El cliente realiza una apertura activa de un puerto y envía una petición de conexión con la dirección IP y el puerto con el que se desea conectar.

Si el servidor está listo responde confirmando la recepción, si no, rechaza la petición.

El cliente confirma la recepción y comienza el envío de datos.



Establecimiento de una conexión TCP

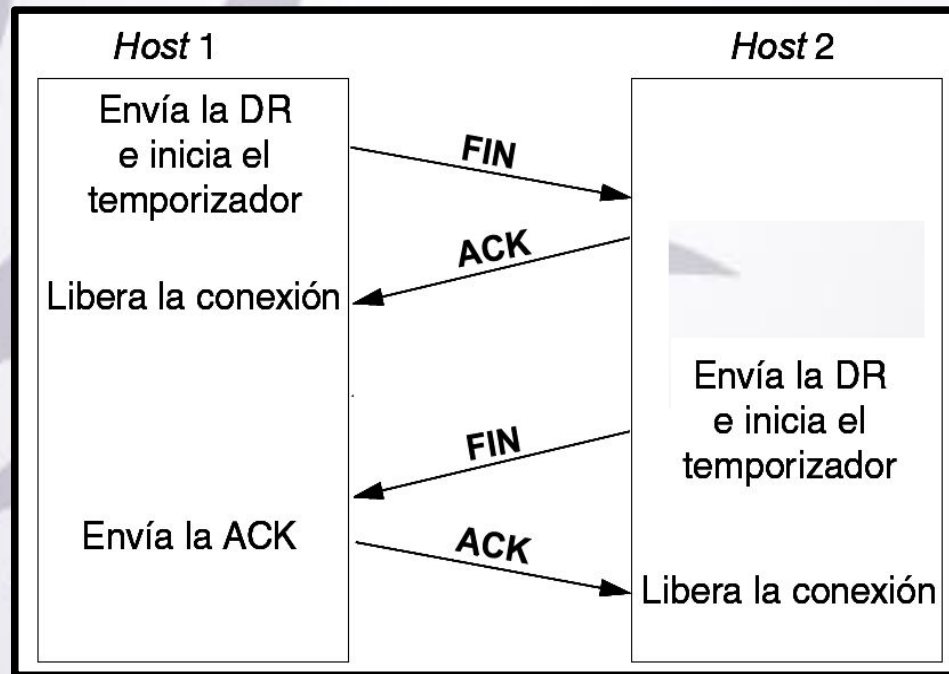


El servidor está a la espera de una conexión. El cliente escoge un número de secuencia 'x' y envía al servidor un segmento con $\text{SYN}=1$. Si el servidor no acepta la conexión responde con un segmento $\text{RST}=1$. Si el servidor acepta la conexión, responde con un segmento $\text{SYN}=1$, $\text{ACK}=1$, confirmando la recepción de 'x' y anunciando su propio número de secuencia inicial. Por último, el cliente confirma la recepción contestando al servidor con un segmento $\text{ACK}=1$.



Liberación de una conexión

La fase de finalización de la conexión usa una negociación en cuatro pasos, terminando la conexión desde cada lado independientemente. Cuando uno de los dos extremos de la conexión desea parar su "mitad" de conexión, transmite un segmento con $FIN=1$, que el otro interlocutor asentirá con un $ACK=1$. Por tanto, una desconexión típica requiere un par de segmentos FIN y ACK desde cada lado de la conexión.





Diferencias entre TCP y UDP

El protocolo UDP

UDP es un protocolo no orientado a conexión. La transferencia de datos es realizada sin haber establecido previamente una conexión con un host destino, y el destinatario recibirá los datos sin enviar una confirmación al host emisor.

El protocolo TCP

TCP está orientado a la conexión. Cuando un host emisor envía datos a una host receptor, el receptor es informado de la llegada de datos y confirma su correcta recepción.



Diferencias entre TCP y UDP

UDP: proporciona un nivel de transporte no fiable de datagramas, ya que apenas añade la información necesaria para la comunicación extremo a extremo. No introduce retardos para establecer una conexión, no mantiene estado de conexión alguno y no realiza seguimiento de estos parámetros. Así, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de sobre TCP. Cuando es más importante la velocidad que la fiabilidad, se utiliza UDP.

TCP: es el protocolo que proporciona un transporte fiable de flujo de bits entre aplicaciones. Está pensado para poder enviar grandes cantidades de información de forma fiable, liberando al programador de gestionar la fiabilidad de la conexión, la cual es responsabilidad de TCP. Pero la complejidad de la gestión de la fiabilidad tiene un coste en eficiencia, añade información a los paquetes para enviar (al menos 20 bytes en TCP y 8 bytes en UDP).

TCP asegura la recepción en destino de la información para transmitir.



FTP (Protocolo de Transferencia de Archivos)

Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El servicio FTP es ofrecido por la capa de aplicación

Utiliza normalmente los puertos de capa de transporte 20 y 21.



FTP (Protocolo de Transferencia de Archivos)

Servidor FTP

Un servidor FTP es un programa que se ejecuta en un equipo servidor normalmente conectado a Internet . Su función es permitir el intercambio de datos entre diferentes servidores/ordenadores.

Cliente FTP

Cuando un navegador no está equipado con la función FTP, o si se quiere cargar archivos en un ordenador remoto, se necesitará utilizar un programa cliente FTP. Algunos clientes FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Microsoft Windows, DOS, GNU/Linux y Unix.



Guía de comandos FTP

Comando	Acción que realiza
open <i>servidor</i>	Inicia una conexión con un servidor FTP.
close o disconnect	Finaliza una conexión FTP sin cerrar el programa cliente.
quit	Finaliza una conexión FTP y la sesión de trabajo con el programa cliente.
delete <i>archivo</i>	Borra un archivo en el servidor.
put <i>archivo</i>	Envía un archivo al directorio activo del servidor.
rename <i>archivo</i>	Cambia el nombre a un archivo en el servidor.
get <i>archivo</i>	Obtiene un archivo.
dir	Muestra el contenido del directorio en el que estamos en el servidor.
mkdir	Crea el directorio indicado de forma remota.



Tanenbaum, Andrew S. *Redes de computadoras, 5ta Edición*. Prentice Hall. 2011. Capítulo 6.5