***Francisco Javier Otero Herrero***

***Grupo ATU***

***20-3-2025***

***Características y conexiones del TCP***

***Contenido***

[Protocolo TCP 2](#_Toc193372867)

[**Principales características del protocolo TCP: 2**](#_Toc193372868)

[**Ejemplo de uso: 3**](#_Toc193372869)

[Proceso de conexión y desconexión con TCP 4](#_Toc193372870)

[**Establecimiento de la Conexión (Three-Way Handshake) 4**](#_Toc193372871)

[**Cierre de la Conexión (Four-Way Handshake) 5**](#_Toc193372872)

[**Diferencias entre la conexión y desconexión 6**](#_Toc193372873)

# Protocolo TCP

El **protocolo TCP (Transmission Control Protocol)** es uno de los principales protocolos de comunicación en redes, utilizado principalmente en Internet para garantizar la entrega fiable de datos entre dispositivos.

Las siglas **TCP/IP** significan Transmission Control Protocol/**Internet Protocol.** El TCP/IP describe una serie de protocolos generales de diseño e implementación para permitir que un equipo pueda comunicarse con una red. Dota a la red de una conectividad “extremo a extremo” especificando cómo se comportan los datos en ella. El mantenimiento de estos protocolos es responsabilidad de la **IETF (Internet Engineering Task Force.**

## Principales características del protocolo TCP:

* Confiabilidad: TCP garantiza que los datos se entregan sin errores, en orden y sin duplicaciones. Para lograrlo, usa mecanismos como la numeración de segmentos y confirmaciones (ACK).
* Orientado a conexión: Antes de transmitir datos, se debe establecer una conexión entre el emisor y el receptor mediante un proceso llamado "Three-Way Handshake" (sincronización de tres pasos).
* Control de flujo: Regula la cantidad de datos que pueden ser enviados sin abrumar al receptor, utilizando un sistema de ventana deslizante (Sliding Window).
* Control de congestión: TCP ajusta dinámicamente la velocidad de transmisión para evitar la congestión en la red y reducir la pérdida de paquetes.
* Segmentación y re ensamblaje: Divide los datos en segmentos más pequeños para su transmisión y los vuelve a ensamblar en el destino en el orden correcto.
* Detección y corrección de errores: Usa un mecanismo de suma de verificación (checksum) para detectar errores en los datos recibidos y solicita retransmisiones si es necesario.
* Entrega ordenada de datos: TCP asegura que los segmentos lleguen en el mismo orden en que fueron enviados. Si un paquete se pierde o llega fuera de orden, se retransmite y reorganiza.
* Multiplexación: Permite que múltiples aplicaciones en un mismo dispositivo usen TCP simultáneamente mediante el uso de puertos (ejemplo: HTTP usa el puerto 80, HTTPS el 443).
* Modo full-duplex: Permite que la comunicación sea bidireccional simultáneamente, lo que significa que ambos extremos pueden enviar y recibir datos al mismo tiempo.
* Protocolo pesado (mayor consumo de recursos): Debido a sus mecanismos de control y fiabilidad, TCP requiere más recursos y ancho de banda en comparación con otros protocolos como UDP.

## Ejemplo de uso:

TCP se utiliza en aplicaciones como:

* ***Navegación web (HTTP/HTTPS)***
* ***Correo electrónico (SMTP, IMAP, POP3)***
* ***Transferencia de archivos (FTP)***
* ***Mensajería instantánea***

# Proceso de conexión y desconexión con TCP

El **proceso de conexión y desconexión en TCP** se basa en un conjunto de pasos bien definidos para garantizar una comunicación confiable entre dos dispositivos en la red:

### Establecimiento de la Conexión (Three-Way Handshake)

Antes de enviar datos, TCP necesita establecer una conexión entre el cliente y el servidor mediante un ***"apretón de manos en tres pasos":***

1. **SYN (Solicitud de conexión)**

* El cliente envía un segmento TCP con el flag SYN (synchronize) activado y un número de secuencia inicial (ISN) aleatorio.
* Esto indica que quiere iniciar una conexión.

1. **SYN-ACK (Confirmación de conexión)**

* El servidor responde con un segmento TCP que tiene SYN + ACK (acknowledge) activados.
* El servidor también elige su propio número de secuencia.

1. **ACK (Confirmación final)**

* El cliente envía un último segmento con el flag ACK activado para confirmar la conexión.
* Ahora, la conexión está establecida y los datos pueden empezar a transmitirse.

### Cierre de la Conexión (Four-Way Handshake)

Para finalizar la conexión de manera segura, TCP usa un proceso de **cuatro pasos:**

1. **FIN (Solicitud de cierre)**

* El dispositivo que quiere cerrar la conexión envía un segmento con el flag FIN (finish) activado.
* Esto indica que ya no tiene más datos para enviar.

1. **ACK (Confirmación de FIN)**

* El otro extremo responde con un ACK para confirmar que recibió la solicitud de cierre.

1. **FIN (Solicitud de cierre del otro lado)**

* Cuando el otro dispositivo termina de enviar sus propios datos, también envía un FIN.

1. **ACK (Confirmación final)**

* El primer dispositivo envía un último ACK para confirmar la terminación de la conexión.
* Luego, la conexión se cierra completamente.

**Después del último ACK**, TCP entra en un estado llamado **TIME\_WAIT**, donde espera un corto período de tiempo antes de liberar completamente la conexión, asegurando que todos los paquetes han sido transmitidos correctamente.

## Diferencias entre la conexión y desconexión

|  |  |  |
| --- | --- | --- |
| ***Características*** | ***Conexión(Three-Way Handsacke)*** | ***Desconexión(Four-Way Handsacke)*** |
| ***Número de pasos*** | 3 pasos (SYNC, SYNC-ACK, ACK) | 4 pasos (FIN, ACK, FIN, ACK) |
| ***Iniciador*** | Cliente inicia la conexión | Cualquiera de los dos puede iniciar el cierre |
| ***Objetivo*** | Establecer una comunicación fiable | Asegurar el cierre ordenado de la conexión |
| ***Uso de FIN*** | No se usa | Se usa para indicar el cierre de cada lado |