

DESCRIPCIÓN GENERAL

- En el **emisor** recibe el mensaje procedente del nivel de aplicación, lo divide en segmentos y los entrega a la capa o nivel de red, asegurándose de que lleguen al otro extremo.
- En el **receptor** se ensamblan todos los segmentos para formar de nuevo el mensaje y entregarlo libre de errores a la capa o nivel de aplicación
- El **nivel de transporte ocurre exclusivamente en los equipos terminales** y no en dispositivos de interconexión que solamente tienen niveles de red, enlace y físico.
- Los protocolos de transporte se implementan en el sistema operativo de los equipos terminales.



DESCRIPCIÓN GENERAL

- La capa de transporte es donde, como su nombre indica, **los datos se transportan de un host a otro**.
- La capa de transporte **no tiene conocimiento** del **tipo de host** de destino, el tipo de **medio** por el que deben viajar los datos, la **ruta** tomada por los datos, la **congestión** en un enlace o el tamaño de la red.
- La capa de transporte utiliza **dos protocolos: TCP y UDP**.
 - **TCP** podría asemejarse a recibir una carta certificada. Tienes que firmar antes de que el transportista de correo te la entregue. Esto ralentiza un poco el proceso, pero el remitente sabe con certeza que recibió la carta y cuándo la recibió.
 - **UDP** es como una carta común con un sello. El remitente no puede estar seguro de que has recibido la carta. Hay momentos en que UDP, como una carta, es el protocolo que se necesita.



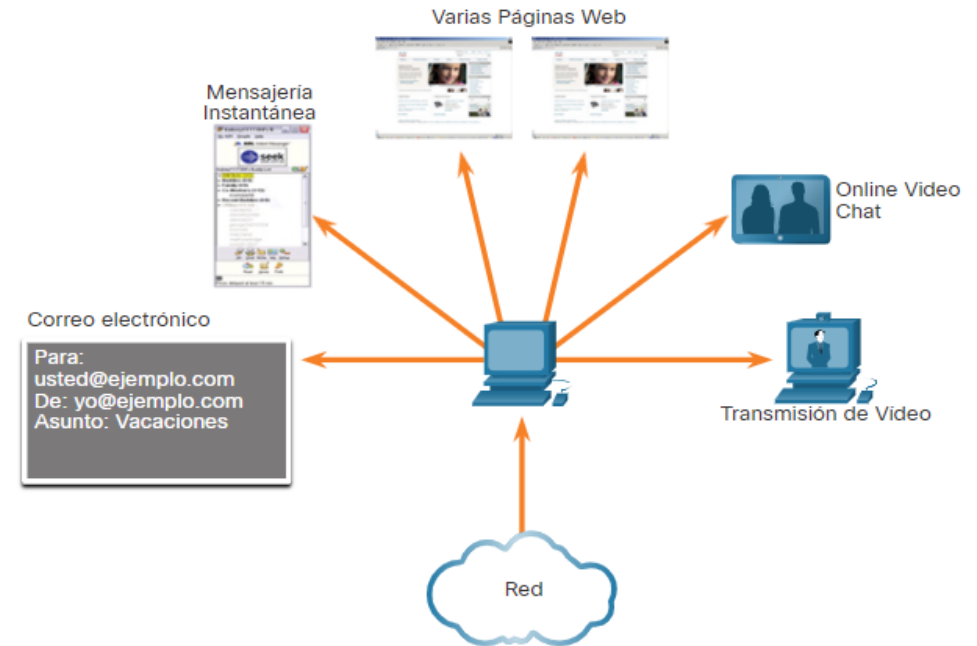
FUNCIONES

- La capa de transporte tiene las siguientes responsabilidades:



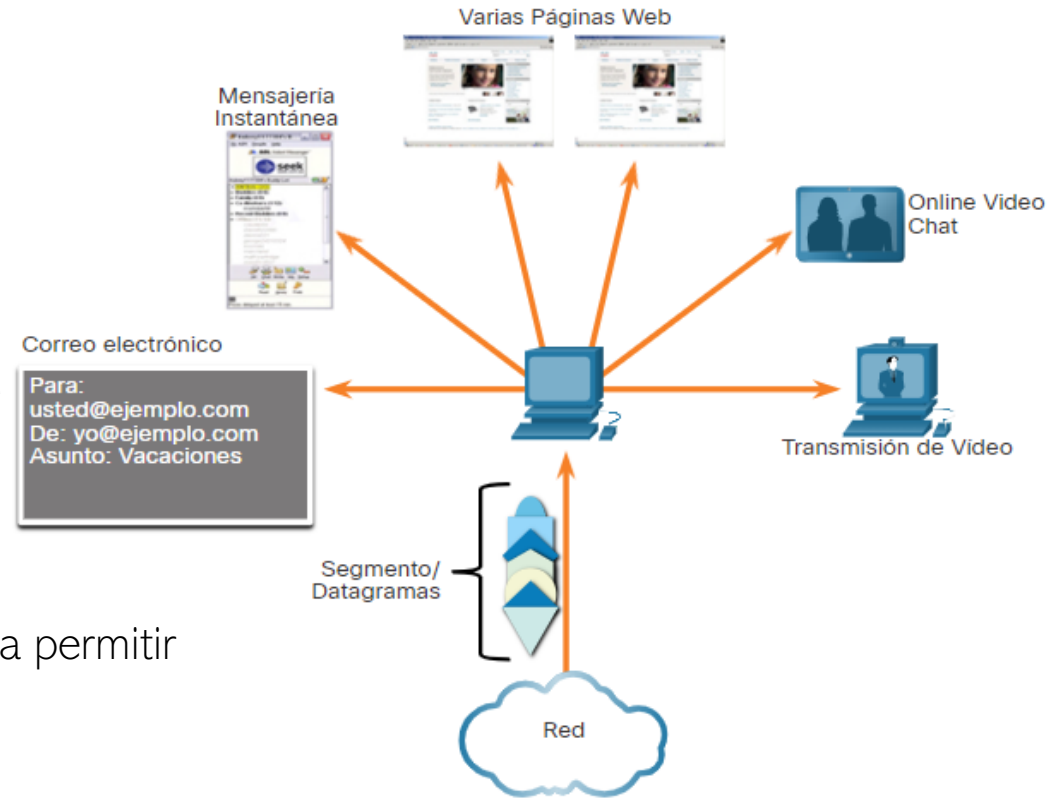
SEGUIMIENTO DE CONVERSACIONES INDIVIDUALES

- Un host puede tener **múltiples aplicaciones** que **se comunican** a través de la red simultáneamente.
- En la capa de transporte, cada conjunto de datos que fluye entre una aplicación de origen y una aplicación de destino se conoce como una **conversación** y se rastrea por separado. Es responsabilidad de la capa de transporte mantener y hacer un seguimiento de todas estas conversaciones.
- La entrega de paquetes se realiza desde un proceso en el equipo emisor, hasta otro proceso en el equipo receptor, usando **puertos**
- Esto permite varias **conexiones simultáneas** en el mismo equipo, por eso es posible abrir varios navegadores y a su vez varias pestañas, todas funcionando a la vez



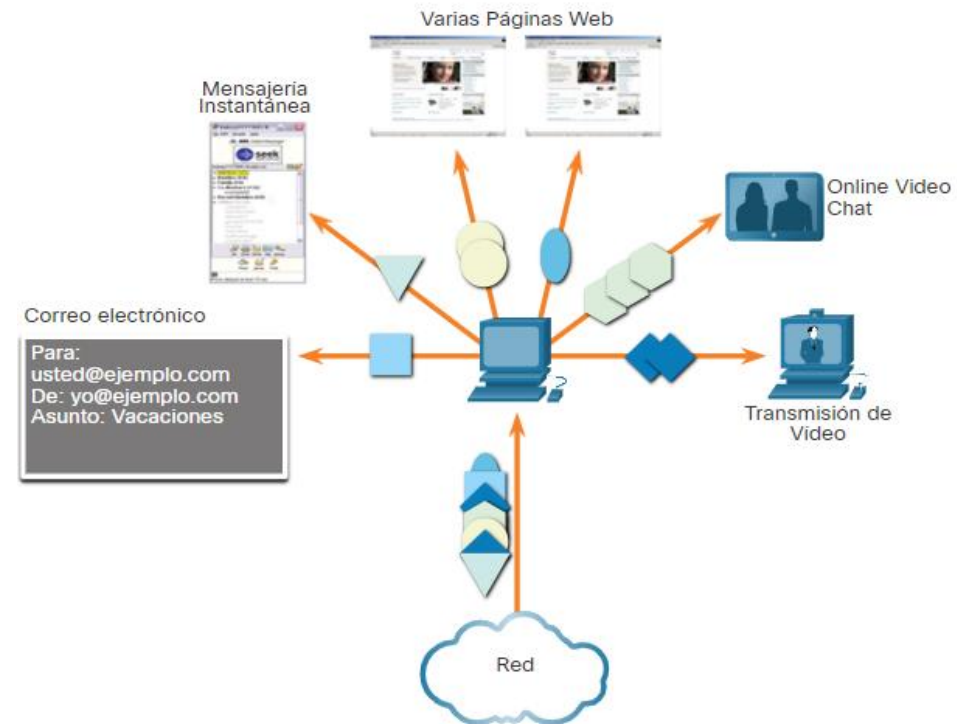
SEGMENTACIÓN DE DATOS Y REARMADO DE SEGMENTOS

- Es responsabilidad de la capa de transporte **dividir los datos** de la aplicación en **bloques de tamaño adecuado**.
- Dependiendo del protocolo de capa de transporte utilizado, los **bloques** de capa de transporte se denominan **segmentos** o **datagramas**.
- Cada segmento lleva un **nº de secuencia** para permitir el ensamblado posterior en el receptor.
- El último segmento lleva un **indicador de fin de secuencia**



AGREGAR INFORMACIÓN DE ENCABEZADO

- El protocolo de capa de transporte también **agrega información de encabezado que contiene datos binarios organizados en varios campos a cada bloque de datos**. Los valores de estos campos permiten que los distintos protocolos de la capa de transporte lleven a cabo variadas funciones de administración de la comunicación de datos.
- Por ejemplo, el host receptor utiliza la **información de encabezado para volver a ensamblar los bloques de datos en un flujo de datos completo para el programa de capa de aplicación de recepción**.
- La capa de transporte garantiza que incluso con múltiples aplicaciones que se ejecutan en un dispositivo, todas las aplicaciones reciben los datos correctos.



IDENTIFICACIÓN DE LAS APLICACIONES

- La capa de transporte debe poder separar y **administrar varias comunicaciones** con diferentes necesidades de requisitos de transporte.
- Para **pasar flujos de datos a las aplicaciones adecuadas**, la capa de transporte identifica la aplicación de destino utilizando un identificador llamado número de **puerto**.
- Como se ilustra en la figura, a cada proceso de software que necesita acceder a la red se le asigna un número de puerto único para ese host.



COMUNICACIONES MÚLTIPLES SEPARADAS

- El nivel de red utiliza solamente direcciones IP para identificar los paquetes que envía a través de la red.
- Sin embargo, **el nivel de transporte añade el puerto**, para distinguir entre los posibles procesos que pueden enviar o recibir datos dentro de un mismo host
- El número de puerto de origen está asociado con la aplicación de origen en el host local, mientras que el número de puerto de destino está asociado con la aplicación de destino en el host remoto.



¿QUÉ ES UN PUERTO?

Un puerto es un número de 16 bits que identifica un proceso local o remoto



¿Cuántos puertos puedo tener?



¿QUÉ ES UN PUERTO?

Un puerto es un número de 16 bits que identifica un proceso local o remoto



- ❑ Como $2^{16} = 65.536$ el rango de puertos va desde 0 a 65.536
- ❑ Estos puertos, a su vez están divididos en tres grupos:
 - ❑ Los puertos bien conocidos
 - ❑ Los puertos registrados
 - ❑ Los puertos dinámicos / privados



LOS PUERTOS

- Total de 65.536 puertos

Puertos bien conocidos (rango del 0 al 1023)

- Servicios de red registrados por [ICANN](#)
- FTP (20 y 21)
- SSH (22)
- TELNET (23)
- SMTP (25)
- DNS (53)
- HTTP (80)
- POP3 (110)
- IMAP (143)
- HTTPS(443)
- ...

Puertos registrados (rango del 1024 al 49151)

- Empleados por aplicaciones de usuario de forma temporal
- Servicios de red registrados por otras entidades
- MS SQL Server (1433)
- Oracle (1525)
- MySQL (3306)
- ...

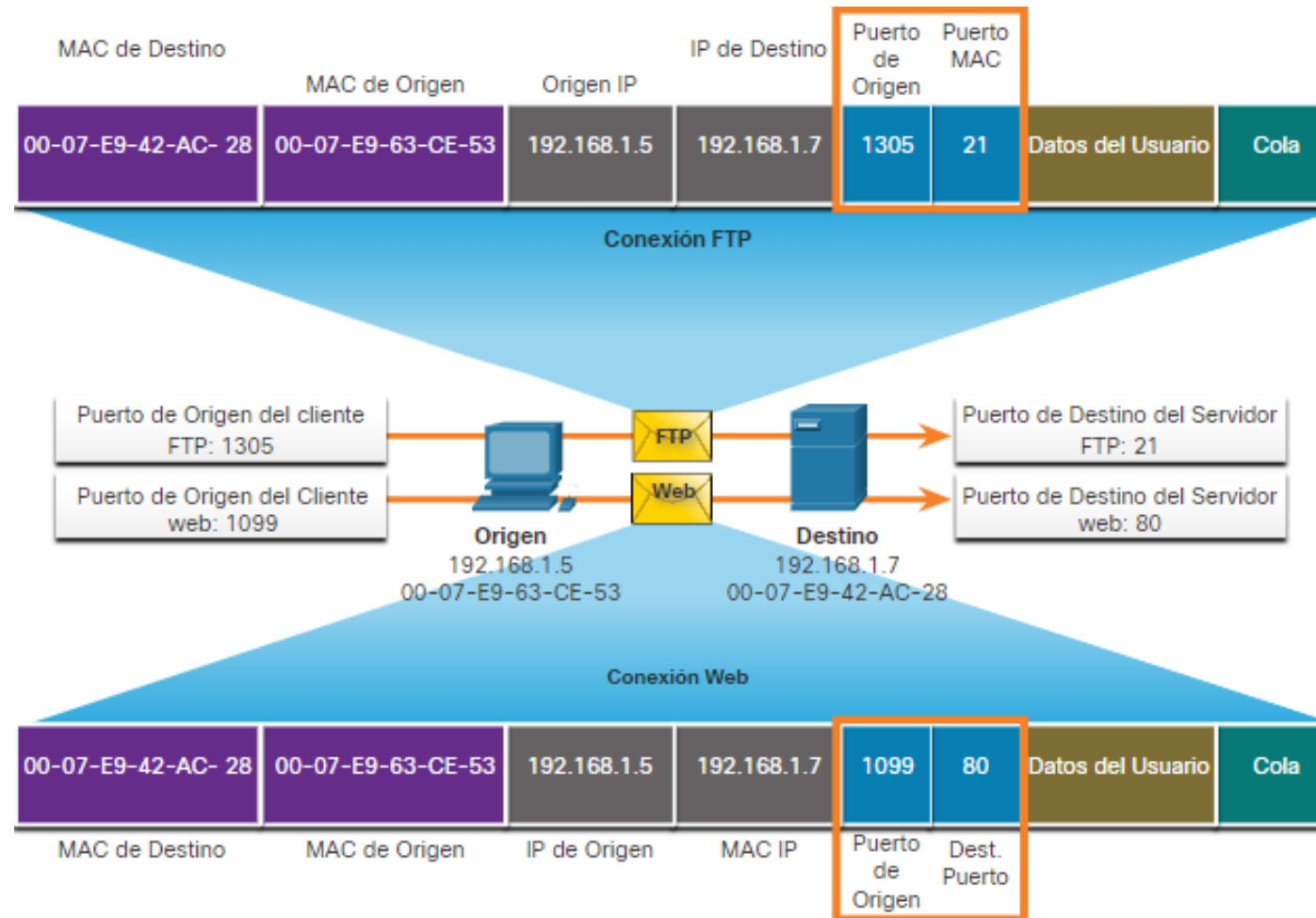
Puertos dinámicos/privados (rango del 49152 al 65535)

- Son empleados por aplicaciones de usuario de forma temporal.
- Cuando se termina comunicación se liberan



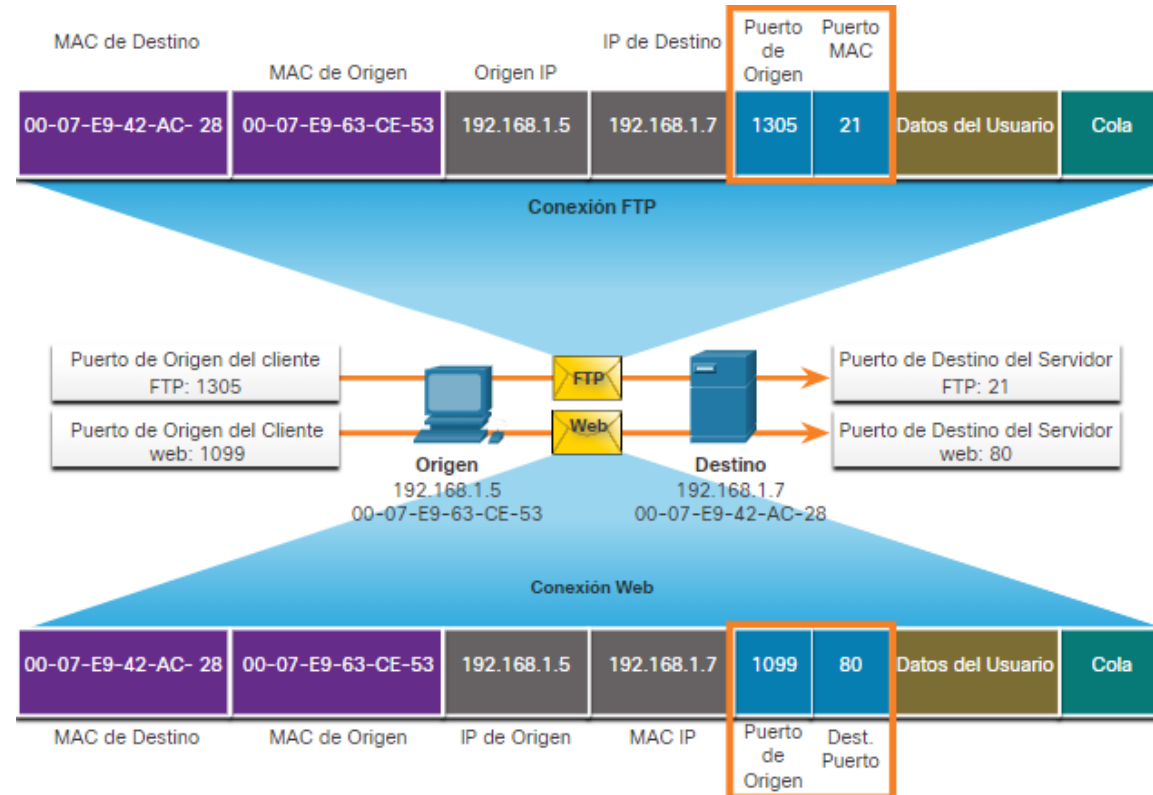
EJEMPLO

- El PC está solicitando simultáneamente servicios FTP y web desde el servidor de destino.



EJEMPLO

- La solicitud FTP generada por el PC incluye las direcciones MAC de Capa 2 y las direcciones IP de Capa 3. La solicitud también identifica el puerto de origen 1305 (es decir, generado dinámicamente por el host) y el puerto de destino, identificando los servicios FTP en el puerto 21.
- El host también ha solicitado una página web del servidor utilizando las mismas direcciones de Capa 2 y Capa 3. Sin embargo, está utilizando el número de puerto de origen 1099 (es decir, generado dinámicamente por el host) y el puerto de destino que identifica el servicio web en el puerto 80.



¿QUÉ ES UN SOCKET?

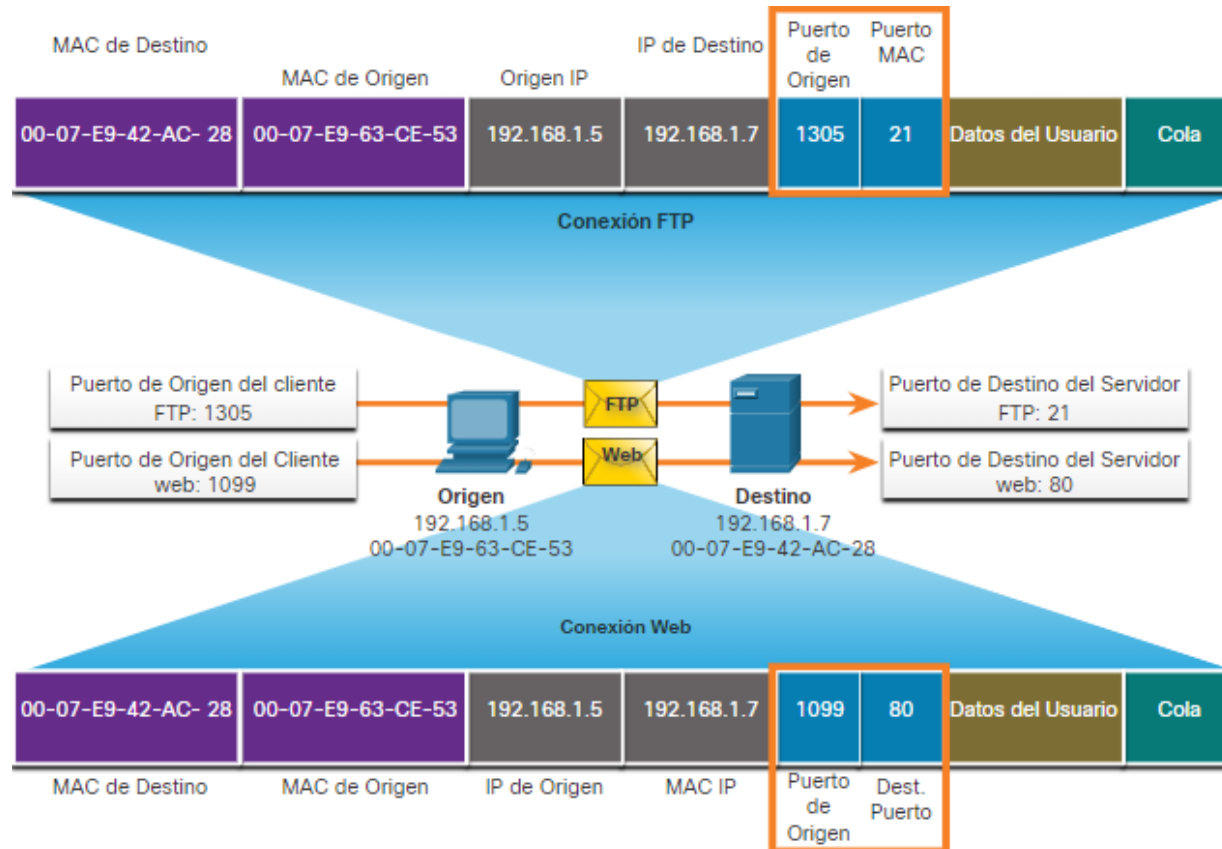
Un socket es un punto de comunicación por el cual un proceso puede emitir o recibir información.

- Un socket se identifica por la pareja “dirección ip: puerto”.
- Ejemplo: 40.67.252.206:443
- Cada socket tiene asociado un espacio de memoria intermedia (buffer) para almacenar los datos que se van a enviar o recibir
- Para establecer la conexión es necesario un socket local y un socket remoto para trabajar de forma conjunta



EJEMPLO

- El socket se utiliza para identificar el servidor y el servicio que solicita el cliente.
- Un socket de cliente (número de puerto de origen): 192.168.1.5:1099
- El socket en un servidor web (número puerto destino): 192.168.1.7:80



Los sockets permiten que los diversos procesos que se ejecutan en un cliente se distingan entre sí. También permiten la diferenciación de diferentes conexiones a un proceso de servidor.



COMANDO NETSTAT

- Para ver las comunicaciones activas en nuestro equipo usamos el comando *netstat*

```
CA: Símbolo del sistema - netstat
Microsoft Windows [Versión 10.0.19044.1586]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>netstat

Conexiones activas

Proto  Dirección local      Dirección remota      Estado
TCP    127.0.0.1:8384        DESKTOP-54DS067:49776 ESTABLISHED
TCP    127.0.0.1:8384        DESKTOP-54DS067:58029 ESTABLISHED
TCP    127.0.0.1:8386        DESKTOP-54DS067:58026 ESTABLISHED
TCP    127.0.0.1:22001       DESKTOP-54DS067:58033 ESTABLISHED
TCP    127.0.0.1:49365       DESKTOP-54DS067:49369 ESTABLISHED
TCP    127.0.0.1:49365       DESKTOP-54DS067:63564 ESTABLISHED
TCP    127.0.0.1:49369       DESKTOP-54DS067:49365 ESTABLISHED
TCP    127.0.0.1:49424       DESKTOP-54DS067:49425 ESTABLISHED
TCP    127.0.0.1:49425       DESKTOP-54DS067:49424 ESTABLISHED
TCP    127.0.0.1:49703       DESKTOP-54DS067:8384  TIME_WAIT
TCP    127.0.0.1:49743       DESKTOP-54DS067:8384  TIME_WAIT
TCP    127.0.0.1:49776       DESKTOP-54DS067:8384  ESTABLISHED
TCP    127.0.0.1:49809       DESKTOP-54DS067:27300 SYN_SENT
TCP    127.0.0.1:58026       DESKTOP-54DS067:8386  ESTABLISHED
TCP    127.0.0.1:58029       DESKTOP-54DS067:8384  ESTABLISHED
TCP    127.0.0.1:58033       DESKTOP-54DS067:22001 ESTABLISHED
TCP    127.0.0.1:63564       DESKTOP-54DS067:49365 ESTABLISHED
TCP    192.168.0.20:49274    a104-126-104-70:http LAST_ACK
TCP    192.168.0.20:49275    a104-126-104-70:http CLOSE_WAIT
```



INTRODUCCIÓN

- TCP está definido en el estándar [RFC 793](#)
- TCP es **orientado a conexión y fiable** → complejo
- Cuando **TCP** recibe datos los fragmenta en trozos (llamados **segmentos**) de hasta 64Kb y los inserta en el área de datos de un datagrama IP
- Como no está garantizada la entrega de esos paquetes, ni el orden, TCP debe encargarse del control y la ordenación, utilizando contadores y números de secuencia



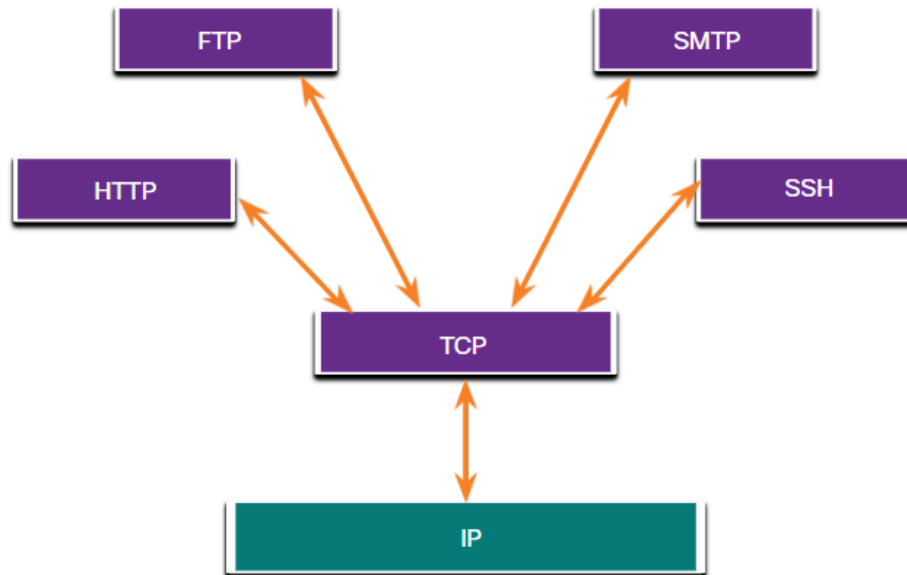
FUNCIONES

- La función del protocolo de transporte TCP es similar al envío de paquetes de los que se hace un **rastreo de origen a destino**. Si se divide un pedido de envío en varios paquetes, un cliente puede verificar en línea para ver el orden de la entrega.
- TCP proporciona confiabilidad y control de flujo mediante estas **operaciones básicas**:
 - Enumerar y rastrear segmentos de datos transmitidos a un host específico desde una aplicación específica
 - Confirmar datos recibidos
 - Retransmitir cualquier información no reconocida después de un cierto período de tiempo
 - Secuenciar datos que pueden llegar en un orden incorrecto
 - Enviar datos a una velocidad eficiente que sea aceptable por el receptor



APLICACIONES QUE UTILIZAN TCP

- TCP maneja todas las tareas asociadas con la división del flujo de datos en segmentos, proporcionando confiabilidad, controlando el flujo de datos y reordenando segmentos.
- TCP libera la aplicación de tener que administrar estas tareas. Las aplicaciones, como las que se muestran en la figura, simplemente puede enviar el flujo de datos a la capa de transporte y utilizar los servicios de TCP.

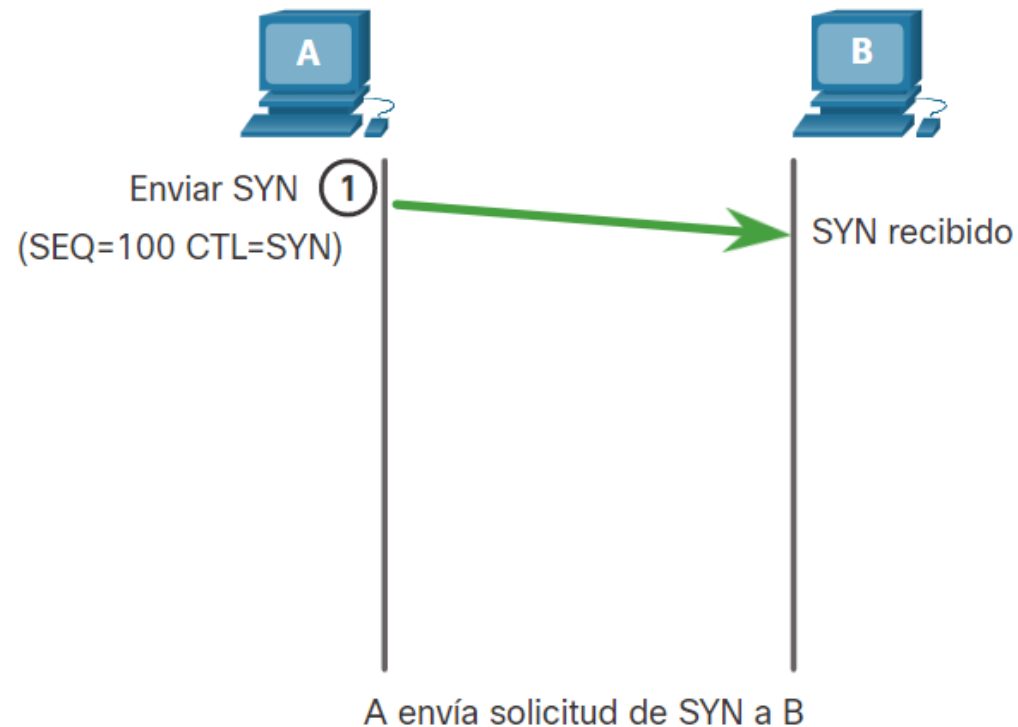


ESTABLECIMIENTO DE LA CONEXIÓN



PASO 1. SYN

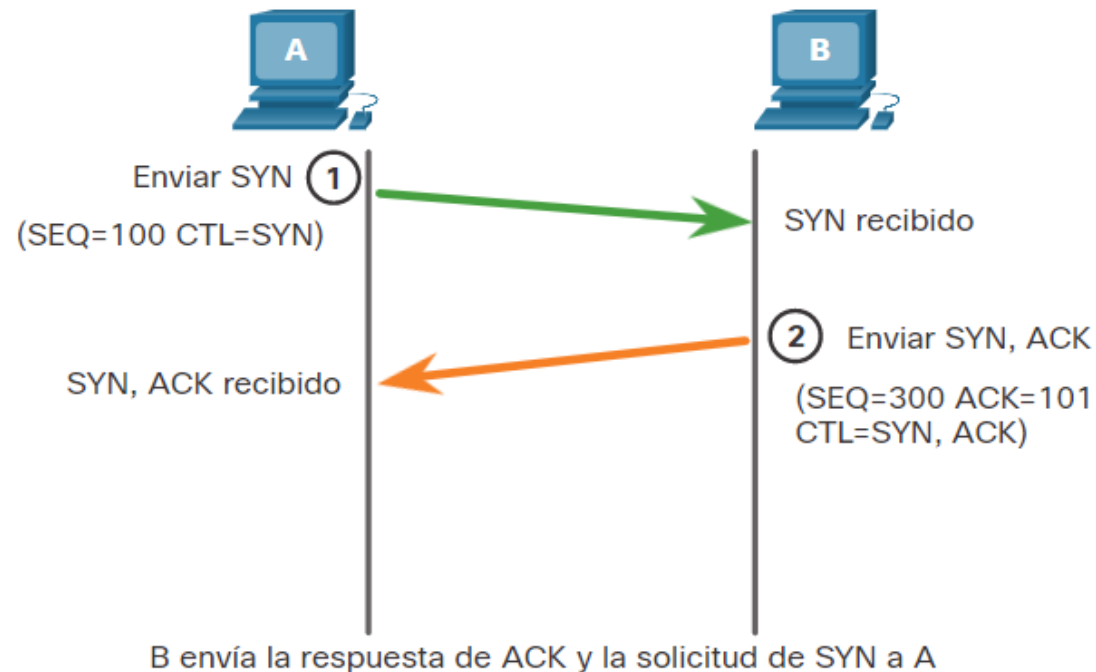
- El cliente solicita una sesión de comunicación con el servidor y realiza la apertura activa de un puerto enviado un paquete **SYN** al servidor



ESTABLECIMIENTO DE LA CONEXIÓN

Paso 2. ACK y SYN

- El servidor comprueba si el puerto está abierto (es decir, si existe un proceso escuchando en dicho puerto)
 - Si no lo está se envía al cliente un paquete de respuesta con el bit RST activado. Rechazo de conexión
 - Si está abierto, se envía un paquete **SYN-ACK**



ESTABLECIMIENTO DE LA CONEXIÓN

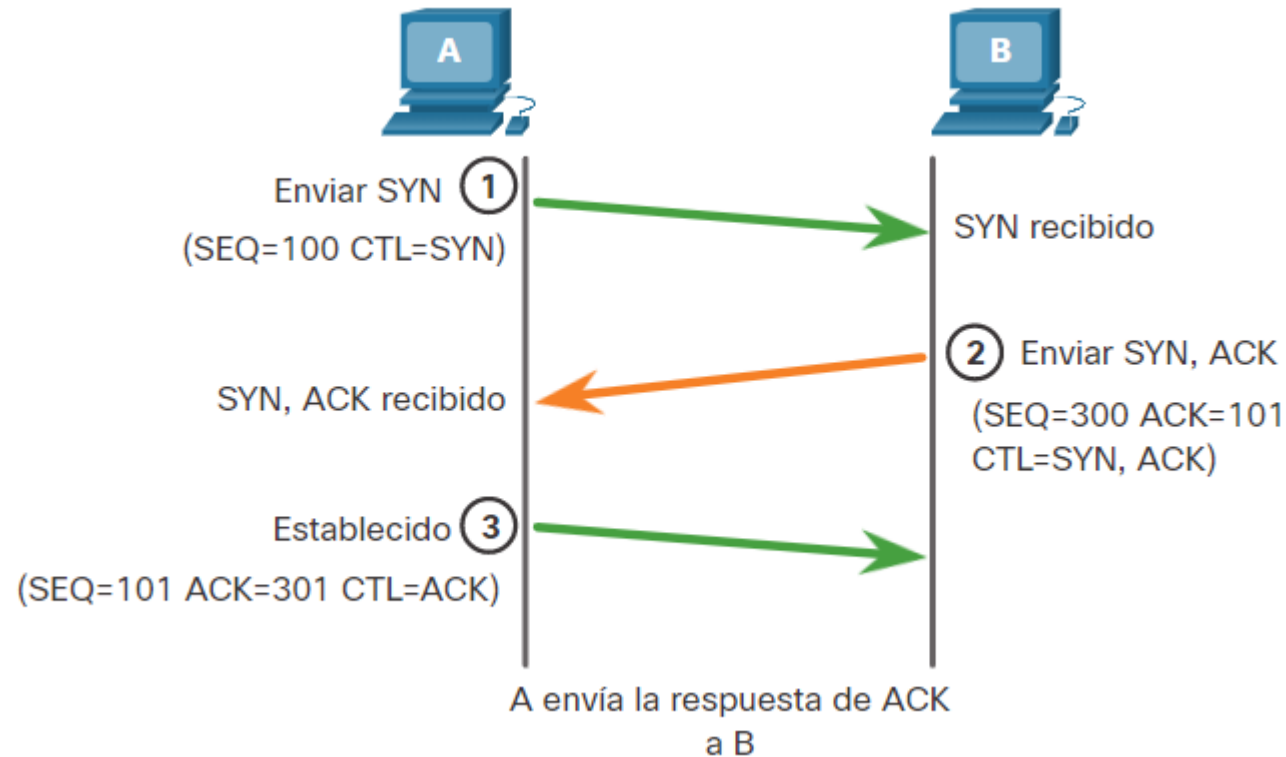
PASO 3. ACK

Establecimiento
de la conexión

Transferencia de
datos

Fin de la
conexión

- El cliente debe responder con un **ACK** completando la negociación en tres pasos



Establecimiento conexión
Negociación en tres pasos



TRANSFERENCIA DE DATOS



- En esta etapa se utilizan una serie de **mecanismos** para asegurar la fiabilidad y robustez del protocolo.
- Entre ellos están:
 - el uso del n° de secuencia para ordenar los segmentos TCP recibidos y detectar paquetes duplicados,
 - *checksum* para detectar errores
 - temporizadores para detectar pérdidas y retrasos.

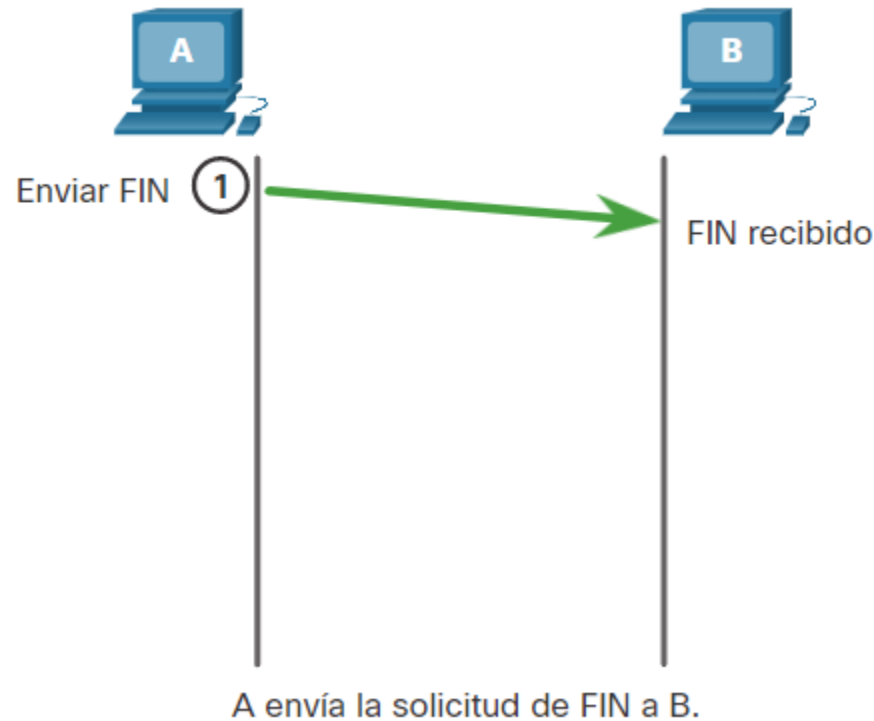


FIN DE LA CONEXIÓN



Paso 1. FIN

- Cuando el cliente no tiene más datos para enviar en la transmisión, envía un segmento con el indicador FIN establecido.

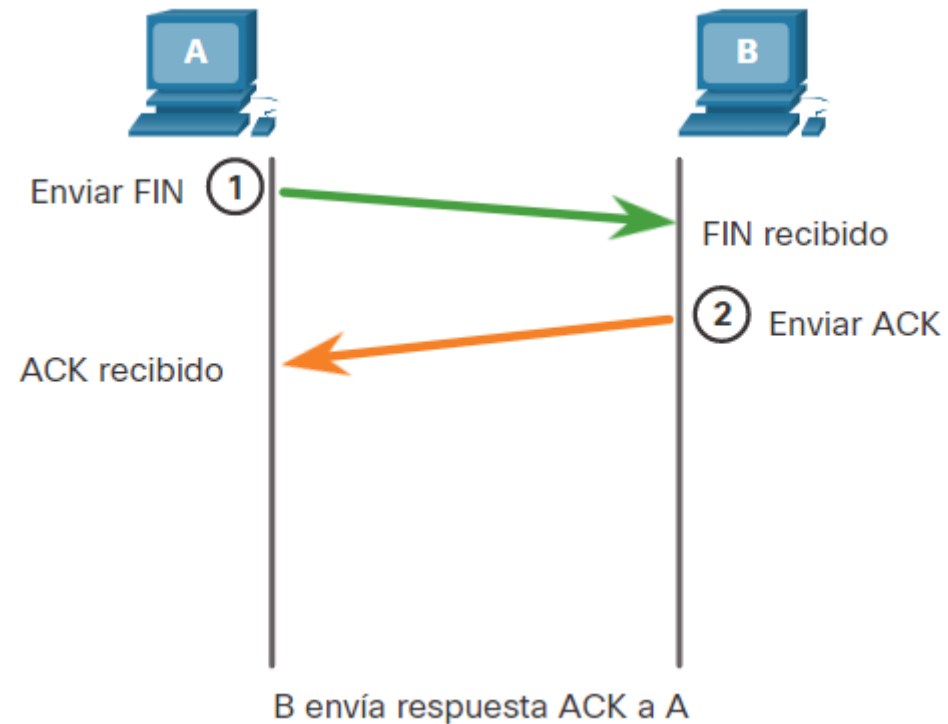


FIN DE LA CONEXIÓN



Paso 2. ACK

El servidor envía un ACK para acusar recibo del FIN para terminar la sesión de cliente a servidor.

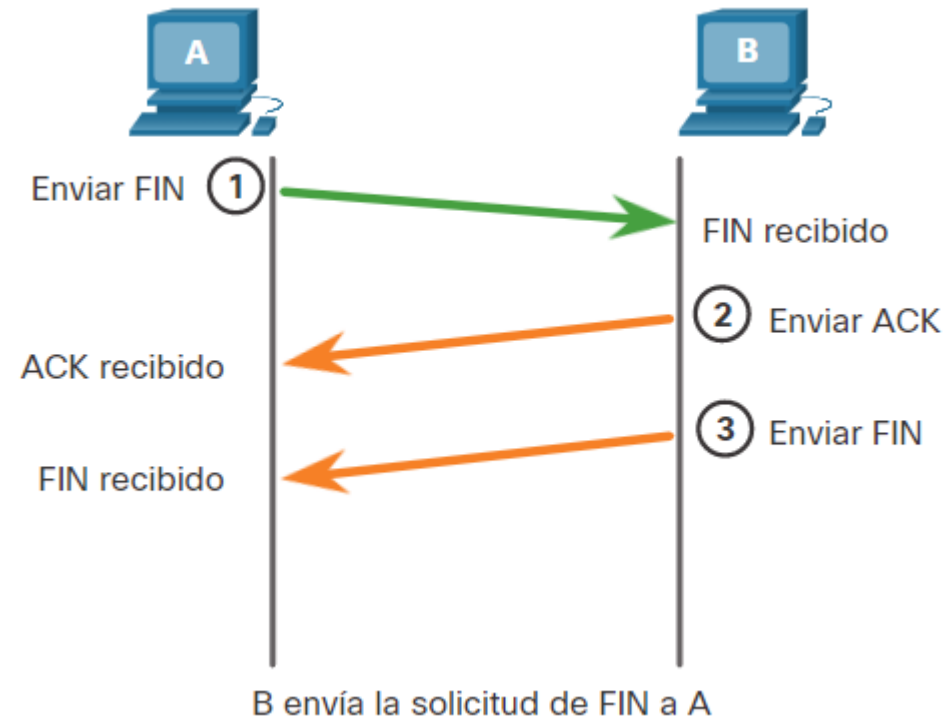


FIN DE LA CONEXIÓN



Paso 3. FIN

El servidor envía un FIN al cliente para terminar la sesión de servidor a cliente.

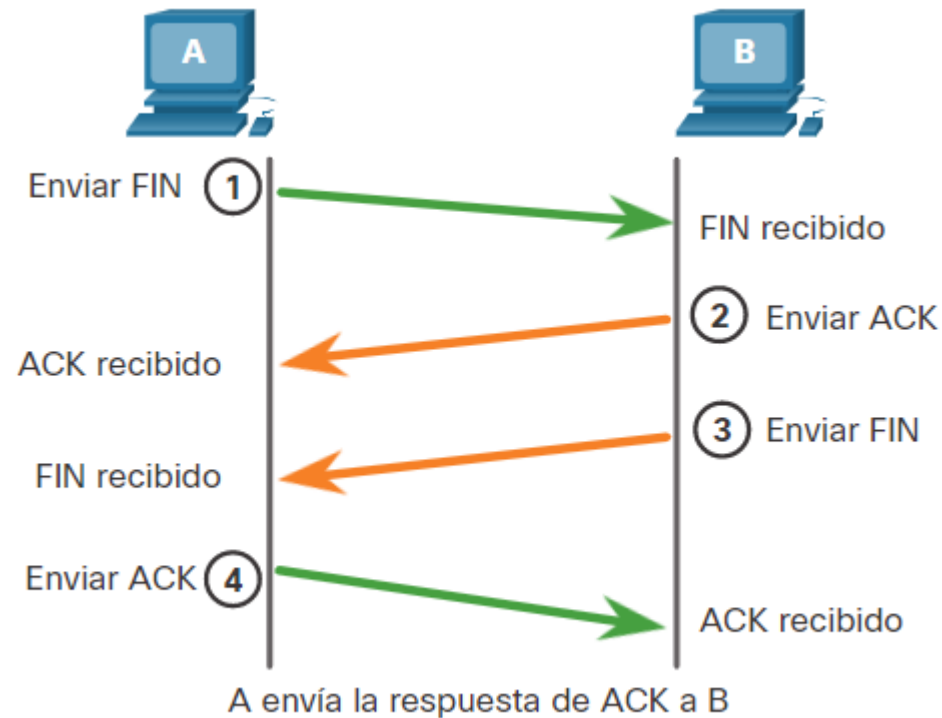


FIN DE LA CONEXIÓN



Paso 4. ACK

El cliente responde con un ACK para dar acuse de recibo del FIN desde el servidor.



Fin conexión
Negociación en cuatro pasos



INTRODUCCIÓN

- UDP está definido en el estándar [RFC 768](#)
- UDP es un protocolo de transporte liviano que ofrece la misma segmentación y rearmado de datos que TCP, pero sin la confiabilidad y el control del flujo de TCP.
- Las características UDP incluyen lo siguiente:
 - Los datos se reconstruyen en el orden en que se recibieron.
 - Los segmentos perdidos no se vuelven a enviar.
 - No hay establecimiento de sesión.
 - El envío no está informado sobre la disponibilidad de recursos.



CARACTERÍSTICAS

- UDP es un protocolo del nivel de transporte basado en el intercambio de **datagramas**
- Es:
 - **No orientado a conexión**
 - Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera
 - **No fiable**
 - No se sabe si los paquetes han llegado correctamente, ya que no hay confirmación de entrega o recepción. No tiene control de flujo, por lo que los paquetes pueden adelantarse unos a otros y llegar desordenados

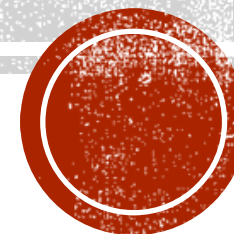


FUNCIONAMIENTO

- Cuando se envían datagramas UDP a un destino, a menudo toman diferentes rutas y llegan en el orden equivocado.
- UDP no realiza un seguimiento de los números de secuencia de la manera en que lo hace TCP. UDP no tiene forma de reordenar datagramas en el orden en que se transmiten
- Por lo tanto, UDP simplemente reensambla los datos en el orden en que se recibieron y los envía a la aplicación. Si la secuencia de datos es importante para la aplicación, esta debe identificar la secuencia adecuada y determinar cómo se deben procesar los datos.



TEMA 13



CAPA DE PRESENTACIÓN Y SESIÓN

FUNCIONES DE LA CAPA DE PRESENTACIÓN

- Dar formato a los datos del dispositivo de origen, o presentarlos, en una forma compatible para que lo reciba el dispositivo de destino.
- Comprimir los datos de forma tal que los pueda descomprimir el dispositivo de destino.
- Cifrar los datos para transmitirlos y descifrarlos al recibirlos.

EJEMPLO

Entre los formatos gráficos de imagen conocidos que se utilizan en redes, se incluyen los siguientes: formato de intercambio de gráficos (GIF), formato del Joint Photographic Experts Group (JPEG) y formato de gráficos de red portátiles (PNG).



CAPA DE PRESENTACIÓN Y SESIÓN

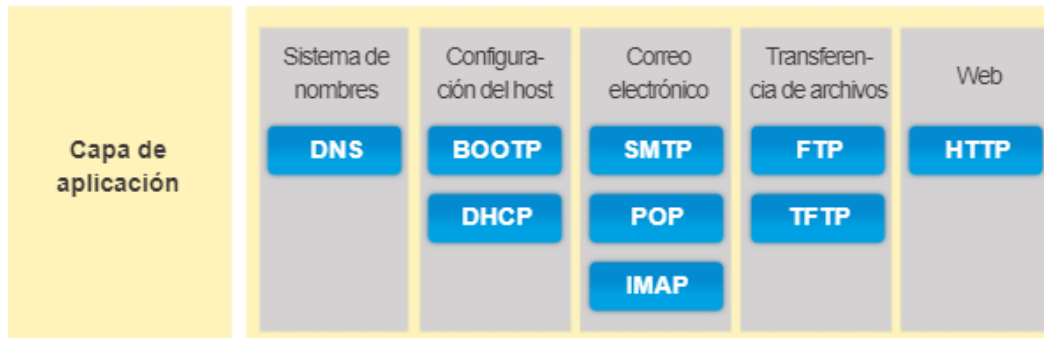
CAPA DE SESIÓN

- Crea y mantiene diálogos entre las aplicaciones de origen y destino.
- maneja el intercambio de información para iniciar los diálogos y mantenerlos activos, y para reiniciar sesiones que se interrumpieron o que estuvieron inactivas durante un período prolongado.



CAPA DE APLICACIÓN

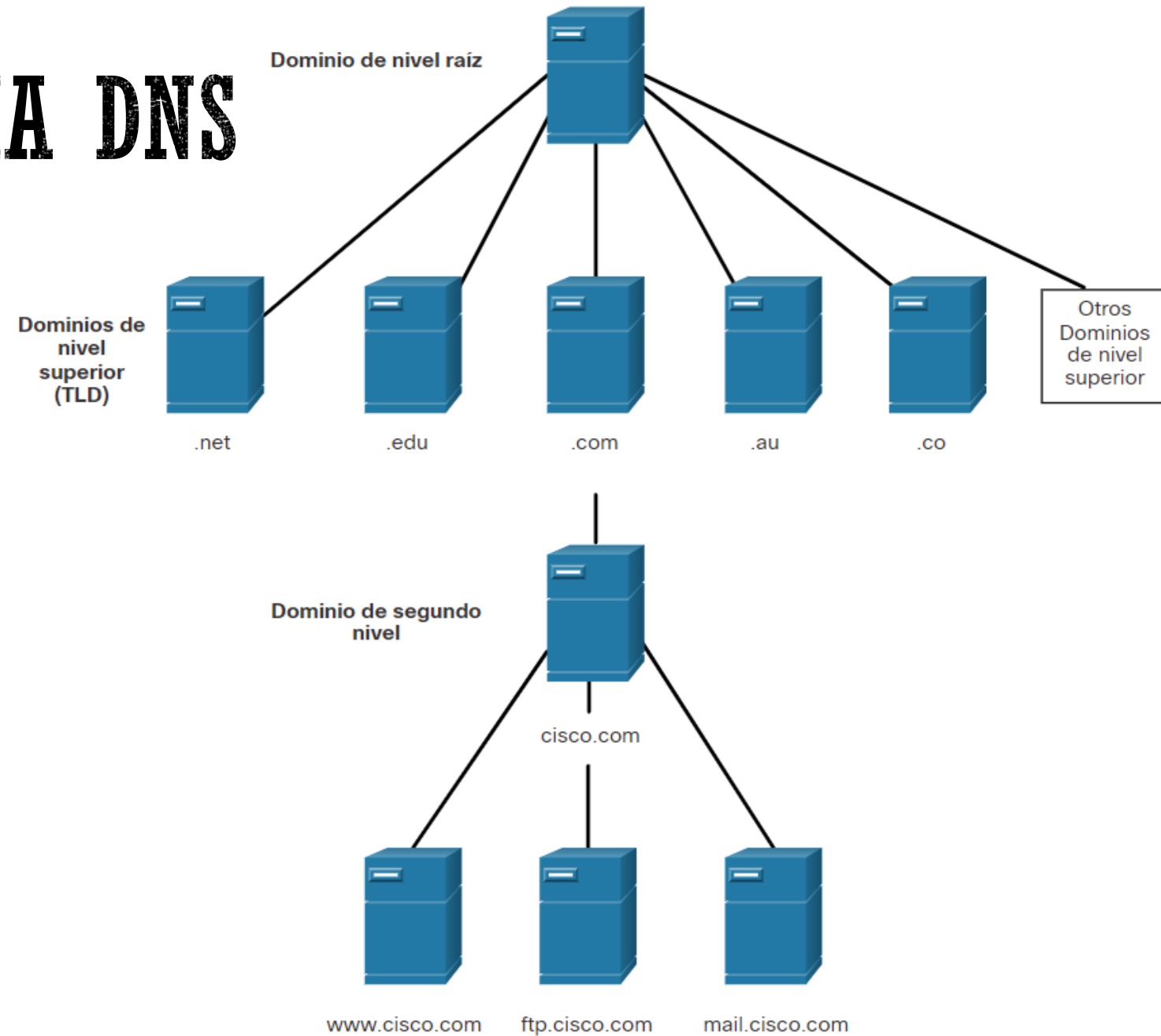
- La función principal es la de proporcionar al usuario servicios de cualquier tipo.
- Principales servicios y sus protocolos asociados



Servicio de red	Protocolos
Resolución de nombres de dominio	DNS
Configuración dinámica de host	DHCP, BOOTP, APIPA...
Transferencia de ficheros	FTP, TFTP
Navegar por la web	HTTP, HTTPS
Correo electrónico	SMTP, POP3, IMAP4
Chat	IRC
Streaming	RTSP
Administración remota	TELNET, SSH, ..
..	..



JERARQUÍA DNS



COMANDO NSLOOKUP

- Los sistemas operativos informáticos también cuentan con una herramienta llamada **nslookup** que permite que el usuario consulte de forma manual los servidores de nombres para resolver un nombre de host dado.
- Esta utilidad también puede utilizarse para solucionar los problemas de resolución de nombres y verificar el estado actual de los servidores de nombres.

```
C:\Users> nslookup
Default Server:  dns-sj.cisco.com
Address:  171.70.168.183
> www.cisco.com
Server:  dns-sj.cisco.com
Address:  171.70.168.183
Name:  origin-www.cisco.com
Addresses:  2001:420:1101:1::a
           173.37.145.84
Aliases:  www.cisco.com
> cisco.netacad.net
Server:  dns-sj.cisco.com
Address:  171.70.168.183
Name:  cisco.netacad.net
Address:  72.163.6.223
>
```

