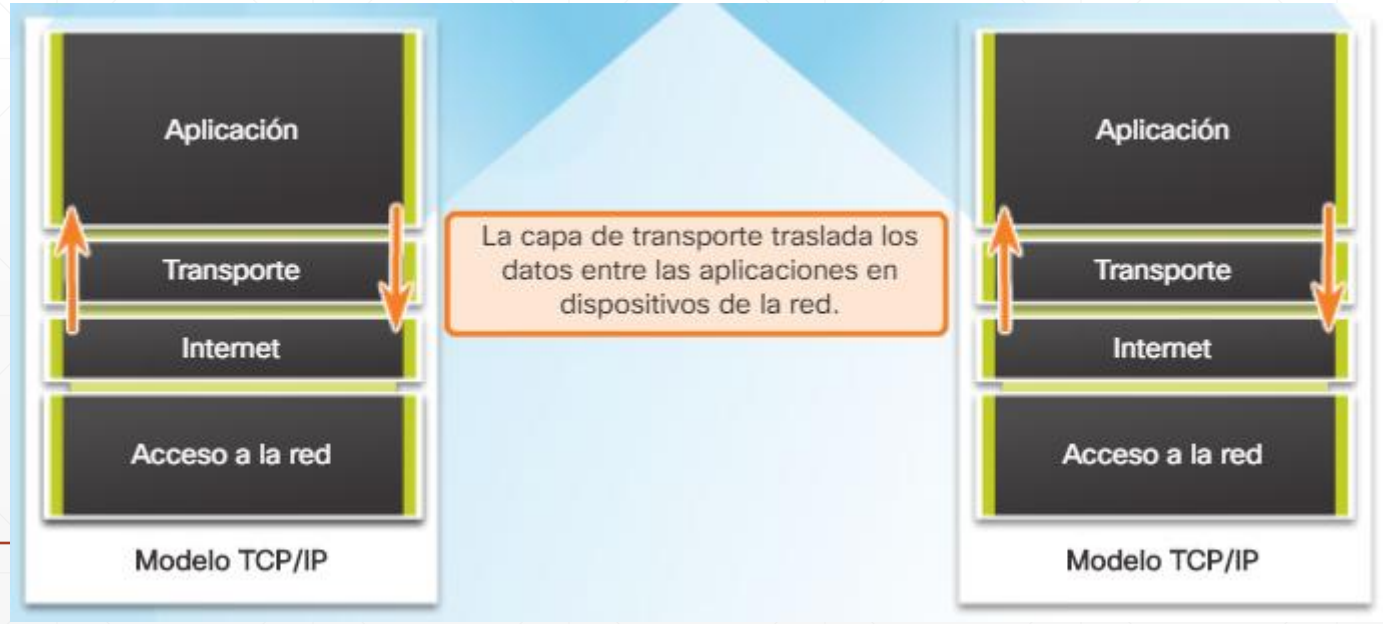
Universidad Rafael Landívar
Tradición Jesuita en Guatemala

Capa 4 - Transporte

Redes I

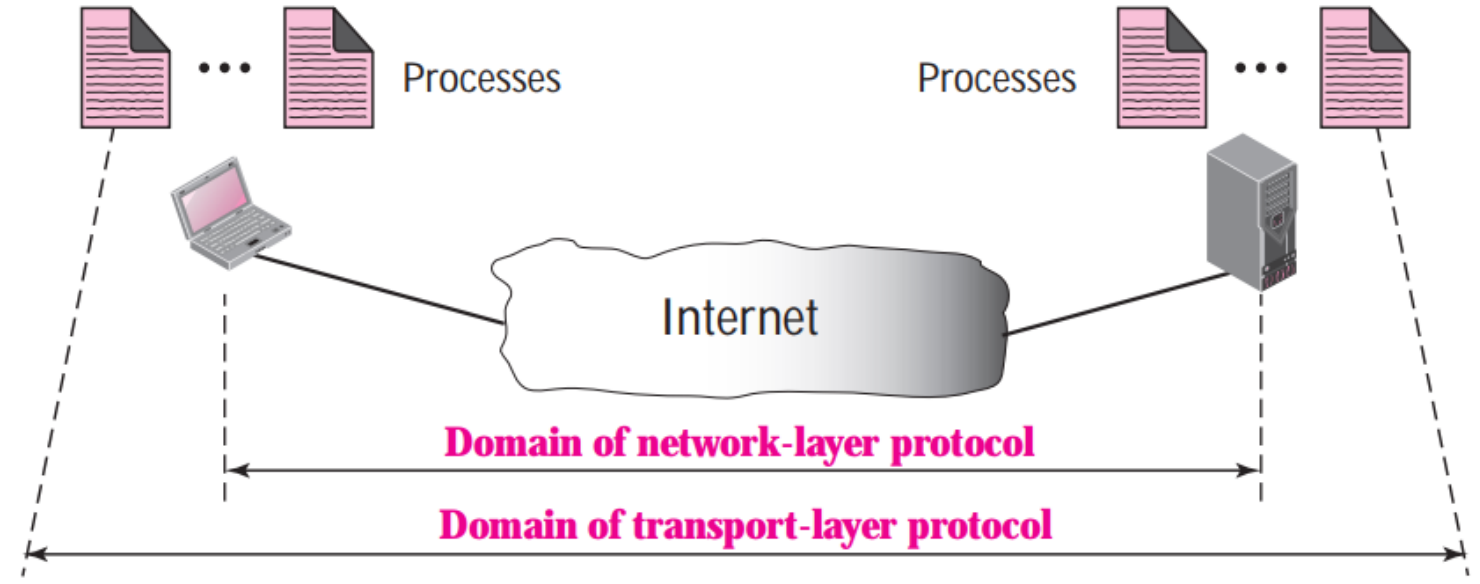
Función de la capa de transporte

- La capa de transporte es responsable de establecer una sesión de comunicación temporal entre dos aplicaciones y de transmitir datos entre ellas.
- Una aplicación genera datos que se envían desde una aplicación en un host de origen a una aplicación en un host de destino.



Process-to-process Communication

- La capa de red es la responsable de la comunicación a nivel host-to-host. Un protocolo de capa de red puede entregar un mensaje únicamente al host destino. Sin embargo esta es una entrega incompleta. El mensaje aun debe ser dirigido al proceso correcto.
- Un protocolo de capa de transporte es responsable de la entrega de un mensaje al proceso apropiado.



Responsabilidad de la capa de transporte

- Seguimiento de conversaciones individuales
- Segmentación de datos y rearmado de segmentos
- Identificación de las aplicaciones

Seguimiento de conversaciones individuales

- Cada conjunto de datos que fluye entre una aplicación de origen y una de destino se conoce como conversación.
- Un host puede tener varias conversaciones en simultaneo a través de la red con otro host.
- Es responsabilidad de la capa de transporte mantener y hacer un seguimiento de todas estas conversaciones.

Segmentación de datos y rearmado de segmentos

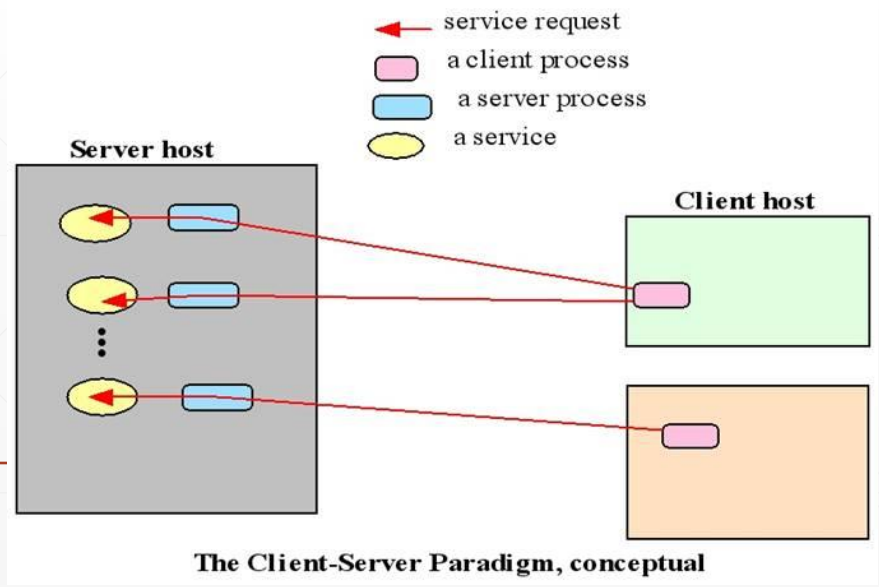
- Se deben preparar los datos para el envío a través de los medios en partes manejables.
- Los protocolos de la capa de transporte tienen servicios que segmentan los datos de aplicación en bloques de un tamaño apropiado.
- Se agrega un encabezado a cada bloque de datos para el rearmado. Este encabezado se utiliza para hacer un seguimiento del flujo de datos.
- En el destino, la capa de transporte debe poder reconstruir las porciones de datos en un flujo de datos completo que sea útil para la capa de aplicación.

Identificación de las aplicaciones

- Para pasar flujos de datos a las aplicaciones adecuadas, la capa de transporte debe identificar la aplicación objetivo.
- La capa de transporte asigna un identificador a cada aplicación, llamado número de **puerto**.
- A todos los procesos de software que requieran acceso a la red se les asigna un número de puerto exclusivo para ese host.

Paradigma cliente-servidor

- Existen pocas maneras para lograr la comunicación proceso-a-proceso, la más común es a través del **paradigma cliente-servidor**.
- Un proceso en el host local, llamado ***cliente***, necesita servicios de un proceso usualmente en un host remoto, llamado ***servidor***.
- Ambos procesos (cliente y servidor) tienen el mismo nombre. Ejemplo: NTP
- Para entablar comunicación se debe definir:
  - Host local
  - Proceso local
  - Host remoto
  - Proceso remoto

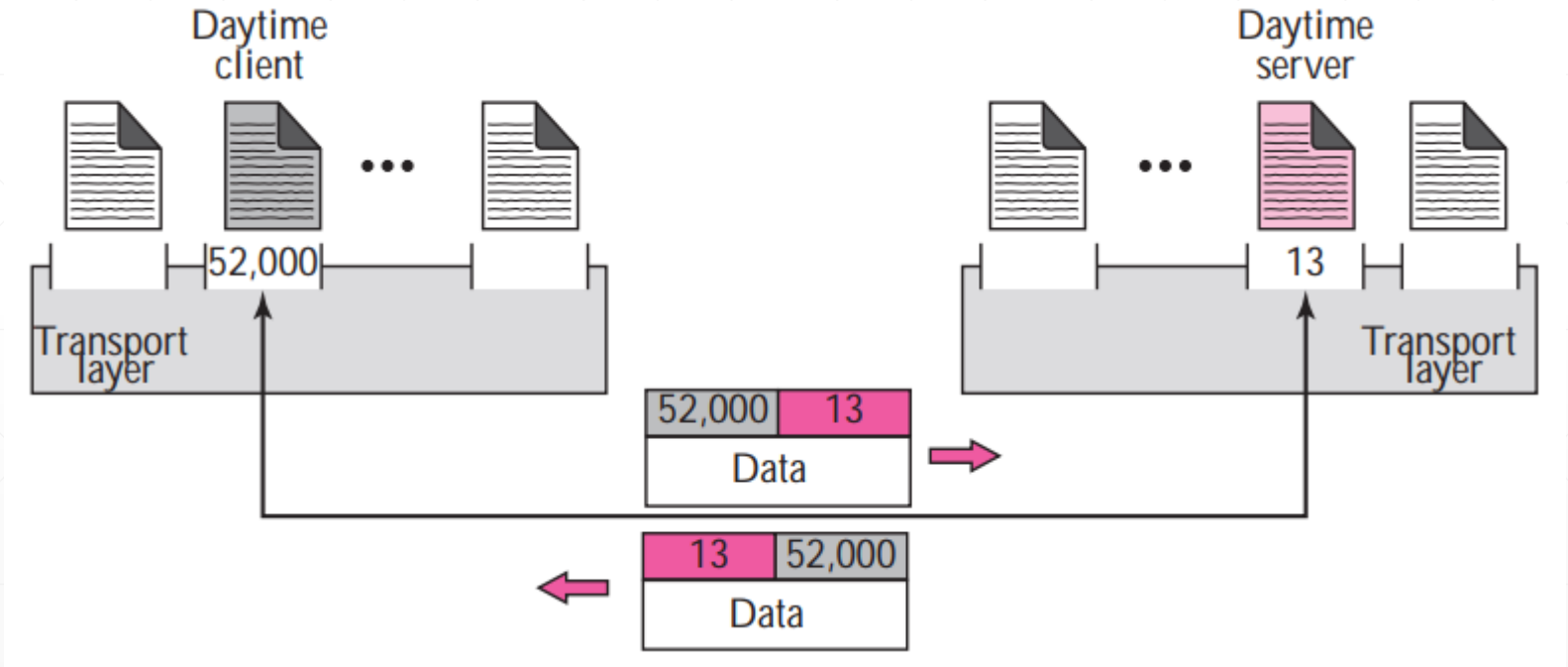


The Client-Server Paradigm, conceptual

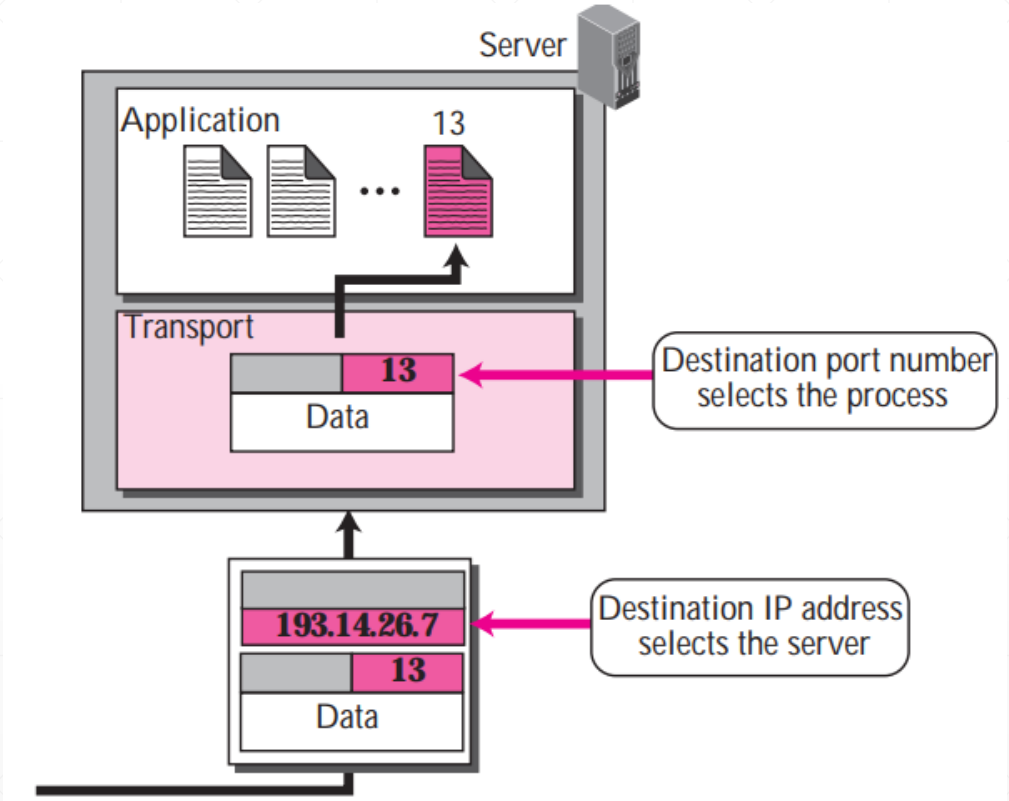
Números de puertos

- El host local y el host remoto se definen utilizando direcciones IP.
- Para definir los procesos, se necesitan identificadores secundarios denominados **números de puerto**.
- En la suite del protocolo TCP/IP, los puertos son números enteros entre **0** y **65,535**.
- El proceso cliente se define a sí mismo con un número de puerto llamado **puerto efímero**. Un puerto efímero generalmente es un número aleatorio mayor a **1,024**.
- El proceso servidor también debe ser definido a través de un puerto. Pero este puerto no se asigna aleatoriamente.
- TCP/IP ha decidido utilizar puertos universales para servidores, llamados **puertos bien conocidos**.

Números de puerto

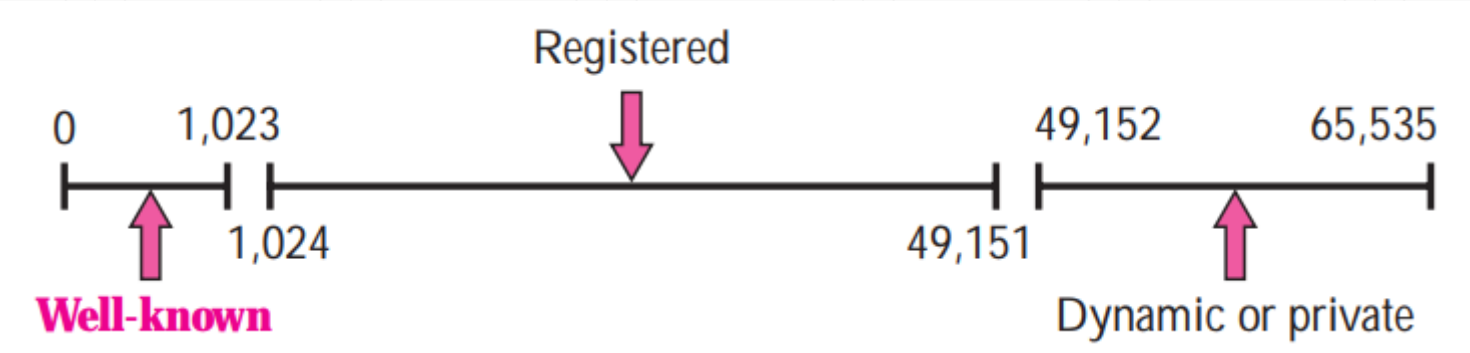


Direcciones IP vs números de puerto



Rangos de puertos de ICANN

- **Puertos bien conocidos**: rango del 0 al 1023 son asignados y controlados por ICANN.
- **Puertos registrados**: rango del 1024 al 49151 no son asignados ni controlados por ICANN. Solo deben de ser registrados en ICANN para evitar duplicados.
- **Puertos dinámicos**: rango del 49152 al 65535 no son ni controlados ni registrados. Son utilizados como puertos temporales o privados. Es recomendado que estos puertos sean asignados como puertos efímeros para las aplicaciones cliente.



Puertos bien conocidos

Tabla 145: Números de puertos bien conocidos (well-known ports) y aplicaciones TCP/IP

Puerto #	TCP / UDP	Keyword	Protocolo (abreviado)	Aplicación o nombre del protocolo / comentarios
7	TCP + UDP	echo	------	Protocolo de eco (echo protocol)
9	TCP + UDP	discard	------	Protocolo Discard
11	TCP + UDP	systat	------	Protocolo Active users
13	TCP + UDP	daytime	------	Protocolo Daytime
17	TCP + UDP	qotd	QOTD	Protocolo Quote of the day
19	TCP + UDP	chargen	------	Protocolo Character Generator
20	TCP	ftp-data	FTP (DATA)	Protocolo ftp (puerto de datos x defecto)
21	TCP	ftp	FTP (CTRL)	Protocolo ftp (control y comandos)
23	TCP	telnet	------	Protocolo telnet
25	TCP	smtp	SMTP	Protocolo SMTP
37	TCP + UDP	time	------	Protocolo Time
43	TCP	nicname	------	Protocolo Whois (también llamado Nicname)
53	TCP + UDP	domain	DNS	Servidor de Nombres de Dominio (Sistema de nombres de Dominio)
67	UDP	bootps	BOOTP / DHCP	Protocolo Bootstrap / Protocolo de configuración automática de hosts (servidor)
68	UDP	bootpc	BOOTP / DHCP	Protocolo Bootstrap / Protocolo de configuración automática de hosts (cliente)
69	UDP	tftp	TFTP	Protocolo para transferencia de archivos triviales (Trivial File Transfer Protocol)
70	TCP	gopher	------	Protocolo Gopher
79	TCP	finger	------	Protocolo Finger para información de usuarios
80	TCP	http	HTTP	Protocolo para Transferencia de Hipertextos (WWW)
110	TCP	pop3	POP	Protocolo de oficina de correos (Post Office Protocol version 3)
119	TCP	nntp	NNTP	Protocolo para transferecias de noticias en red (Network News transfer Protocol)
123	UDP	ntp	NTP	Protocolo de tiempo (Network Time Protocol)
137	TCP + UDP	netbios-ns	------	Protocolo NetBIOS (servicio de nombres)
138	UDP	netbios-dgm	------	Protocolo NetBIOS (servicio de datagramas)
139	TCP	netbios-ssn	------	Protocolo NetBIOS (servicio de sesiones)
143	TCP	imap	IMAP	Protocolo para acceso de mensajes en Internet (Internet Message Access Protocol)
161	UDP	snmp	SNMP	Protocolo Simple para administracion de redes (Simple Network Management Protocol)
162	UDP	snmptrap	SNMP	Simple Network Management Protocol (trap)
179	TCP	bgp	BGP	Protocolo Border gateway
194	TCP	irc	IRC	Protocolo Internet Relay Chat
443	TCP	https	HTTP over SSL	Protocolo HTTP sobre capas seguras (Secure Sockets Layer)
500	UDP	isakmp	IKE	IPSec Internet Key Exchange
520	UDP	router	RIP	Protocolo Para informacion de rutas (Routing Information Protocol (RIP-1 and RIP-2))
521	UDP	ripng	RIPng	Protocolo Para informacion de rutas (neva generación)

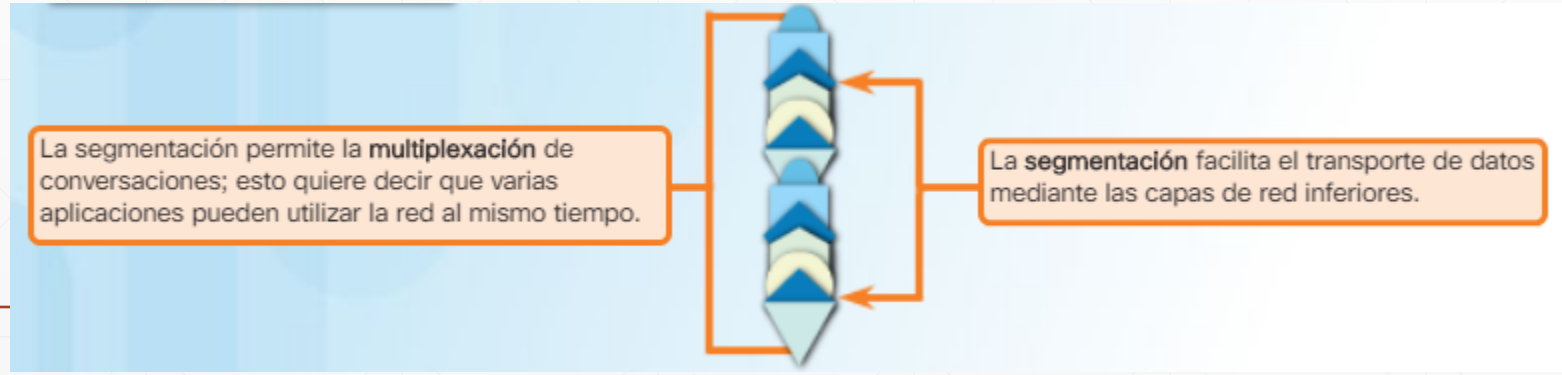
Puertos registrados

Tabla 146: Numeros comunes de puertos / aplicaciones registrados.

Puerto #	TCP / UDP	Keyword	Protocolo (abreviado)	Aplicación o nombre del protocolo / comentarios
1512	TCP + UDP	wins	WINS	Microsoft Windows Internet Naming Service
1701	UDP	l2tp	L2TP	Layer Two Tunneling Protocol
1723	TCP	pptp	PPTP	Point-To-Point Tunneling Protocol
2049	TCP + UDP	nfs	NFS	Network File System
6000 - 6063	TCP	x11	X11	X Window System

Multiplexación de conversaciones

- La segmentación de los datos en partes más pequeñas permite que se entrelacen (multiplexen) varias comunicaciones de distintos usuarios en la misma red.
- Para identificar cada segmento de datos, la capa de transporte agrega un encabezado que contiene datos binarios organizados en varios campos. 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.



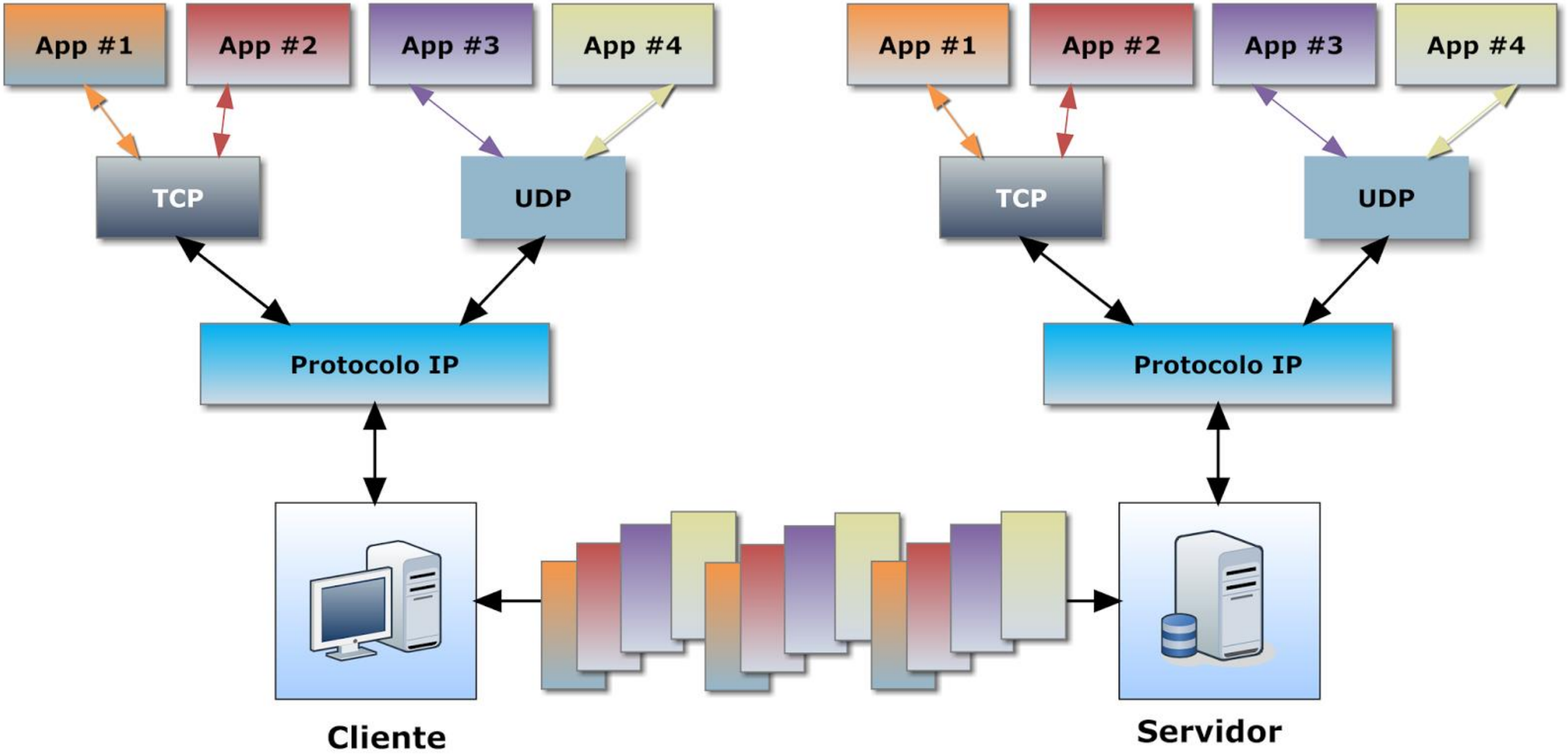


Figura 197: Multiplexación / demultiplexación de procesos en TCP/IP

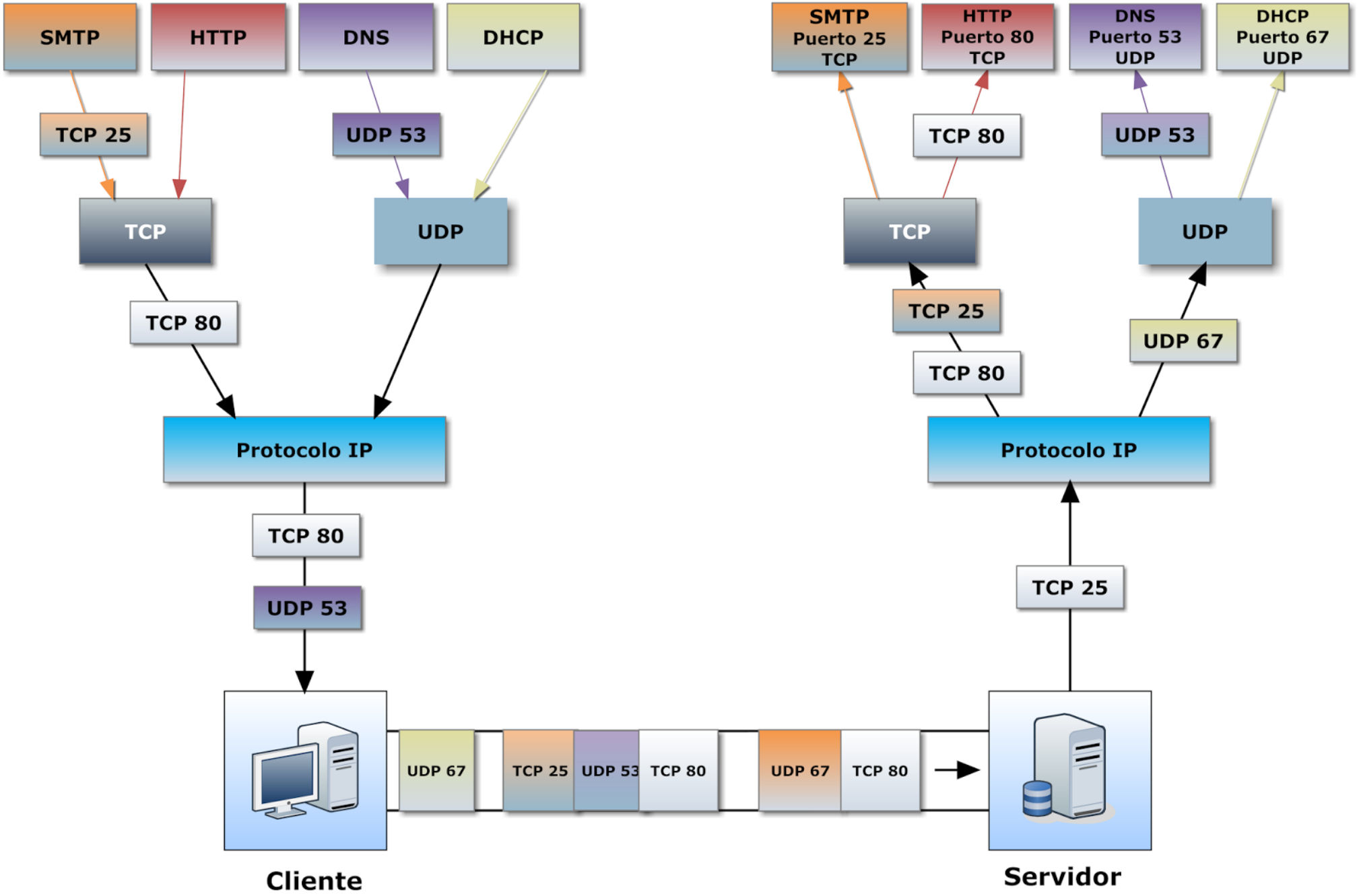
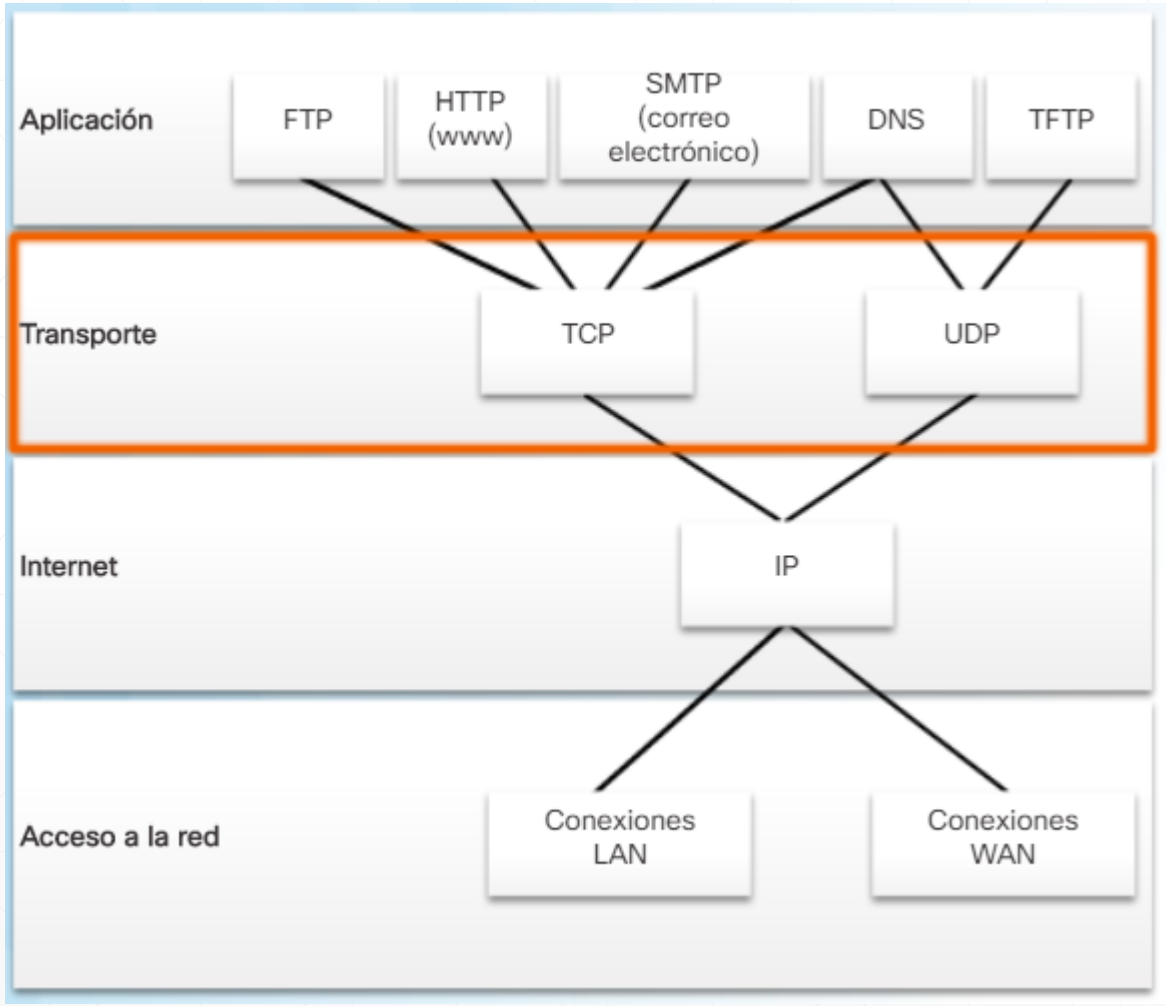


Figura 198: Multiplexación / demultiplexación de procesos en TCP/IP usando puertos TCP/UDP

Confiabilidad de la capa de transporte

- La capa de transporte también es responsable de administrar los requisitos de confiabilidad de las conversaciones. Las diferentes aplicaciones tienen diferentes requisitos de confiabilidad de transporte.
- TCP/IP proporciona dos protocolos de la capa de transporte:
  - TCP: Transmission Control Protocol
  - UDP: User Datagram Protocol
- TCP se considera un protocolo de la capa de transporte confiable y completo, ya que garantiza que todos los datos lleguen al destino. Sin embargo, esto requiere campos adicionales en el encabezado TCP que aumentan el tamaño del paquete y también la demora.
- En cambio, UDP es un protocolo de capa de transporte más simple, aunque no proporciona confiabilidad. Por lo tanto, tiene menos campos y es más rápido que TCP.

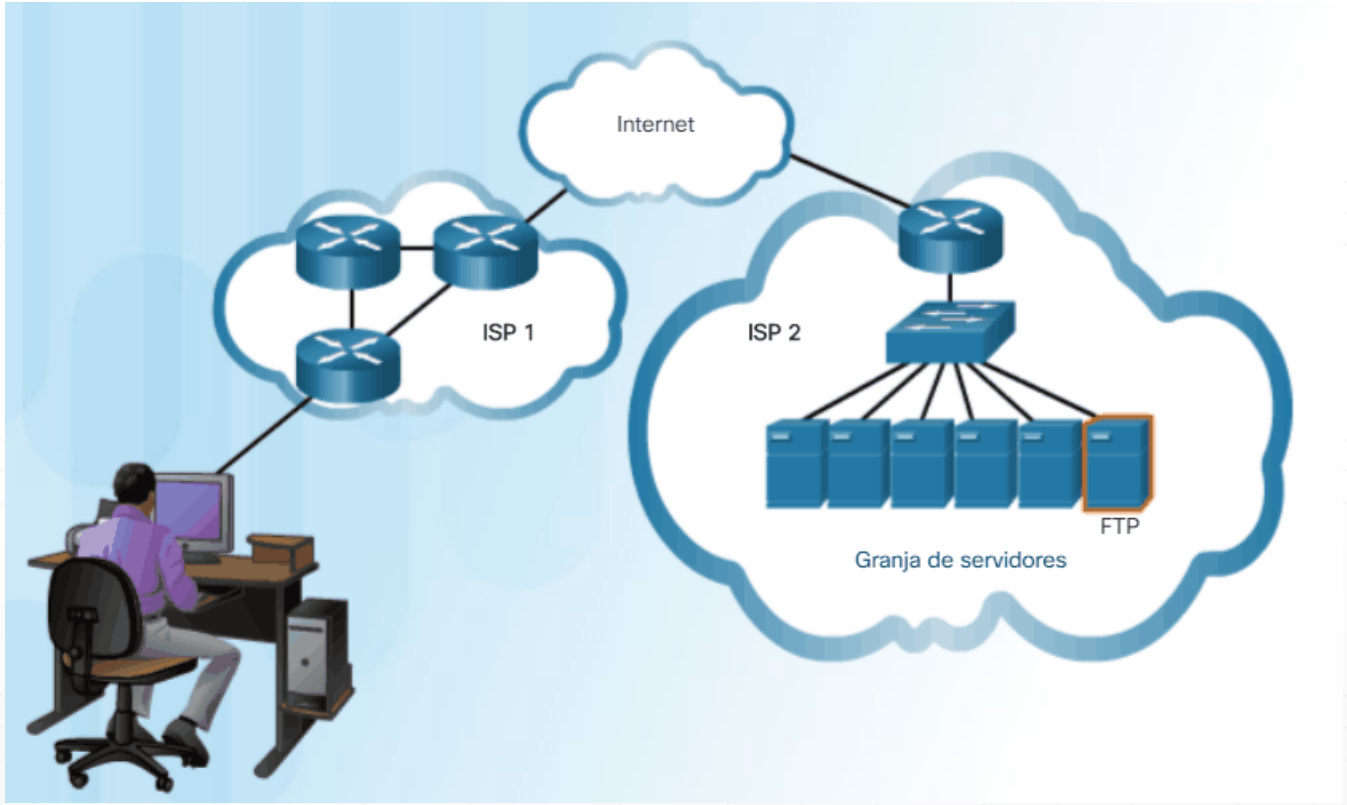

Aplicación
FTP
HTTP (www)
SMTP (correo electrónico)
DNS
TFTP
Transporte
TCP
UDP
Internet
IP
Acceso a la red
Conexiones LAN
Conexiones WAN

TCP

Con TCP, hay tres operaciones básicas de confiabilidad:

- Numeración y seguimiento de los segmentos de datos transmitidos a un host específico desde una aplicación específica
- Reconocimiento de los datos recibidos
- Retransmisión de los datos sin reconocimiento después de un tiempo determinado

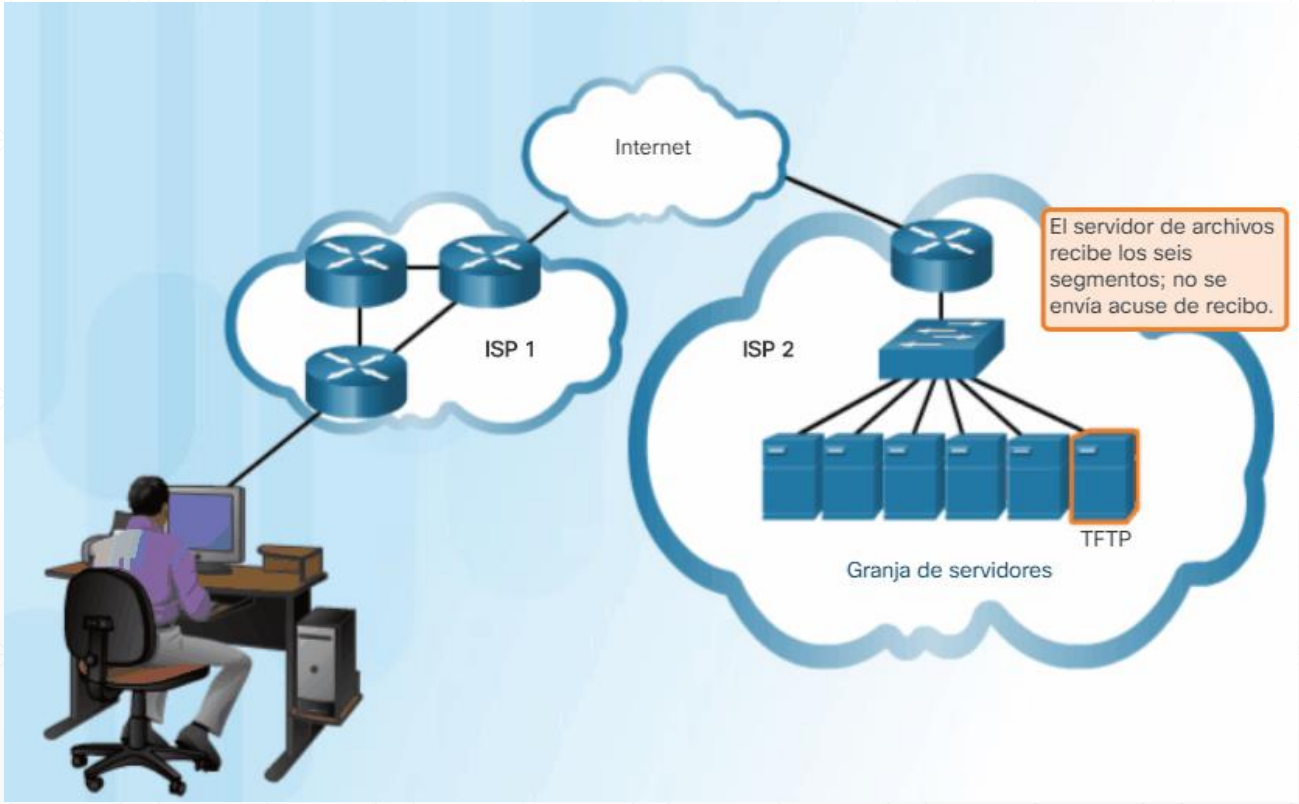
TCP



UDP

- Agregar sobrecarga para garantizar la confiabilidad para algunas aplicaciones podría reducir la utilidad a la aplicación e incluso ser perjudicial. En estos casos, UDP es un protocolo de transporte mejor.
- UDP proporciona las funciones básicas para entregar segmentos de datos entre las aplicaciones adecuadas, con muy poca sobrecarga y revisión de datos. UDP se conoce como un protocolo de entrega de **máximo esfuerzo.**
- En el contexto de redes, la entrega de máximo esfuerzo se denomina “**poco confiable**” porque no hay reconocimiento que indique que los datos se recibieron en el destino.

UDP


Internet
ISP 1
ISP 2
El servidor de archivos recibe los seis segmentos; no se envía acuse de recibo.
TFTP
Granja de servidores

UDP vs TCP

UDP

Telefonía IP

Transmisión de vídeo en vivo

Propiedades de protocolo requeridas:

- Rápido
- Baja sobrecarga
- No requiere reconocimiento
- No reenvía los datos perdidos
- Entrega los datos a medida que llegan

TCP

SMTP/POP (correo electrónico)

HTTP

Propiedades de protocolo requeridas:

- Confiable
- Reconoce los datos
- Reenvía los datos perdidos
- Entrega los datos en orden secuencial

Características TCP

Establecimiento de una sesión

- TCP es un protocolo orientado a la conexión.
- Un protocolo orientado a la conexión es uno que negocia y establece una conexión (o sesión) permanente entre los dispositivos de origen y de destino antes de reenviar tráfico.
- Mediante el establecimiento de sesión, los dispositivos negocian la cantidad de tráfico que se puede reenviar en un momento determinado, y los datos que se comunican entre ambos se pueden administrar detenidamente.

Características TCP

Entrega confiable

- En términos de redes, la confiabilidad significa asegurar que cada segmento que envía el origen llegue al destino. Por varias razones, es posible que un segmento se dañe o se pierda por completo a medida que se transmite en la red.

Entrega en el mismo orden

- Los datos pueden llegar en el orden equivocado, debido a que las redes pueden proporcionar varias rutas que pueden tener diferentes velocidades de transmisión.
- Al numerar y secuenciar los segmentos, TCP puede asegurar que estos se rearmen en el orden correcto.

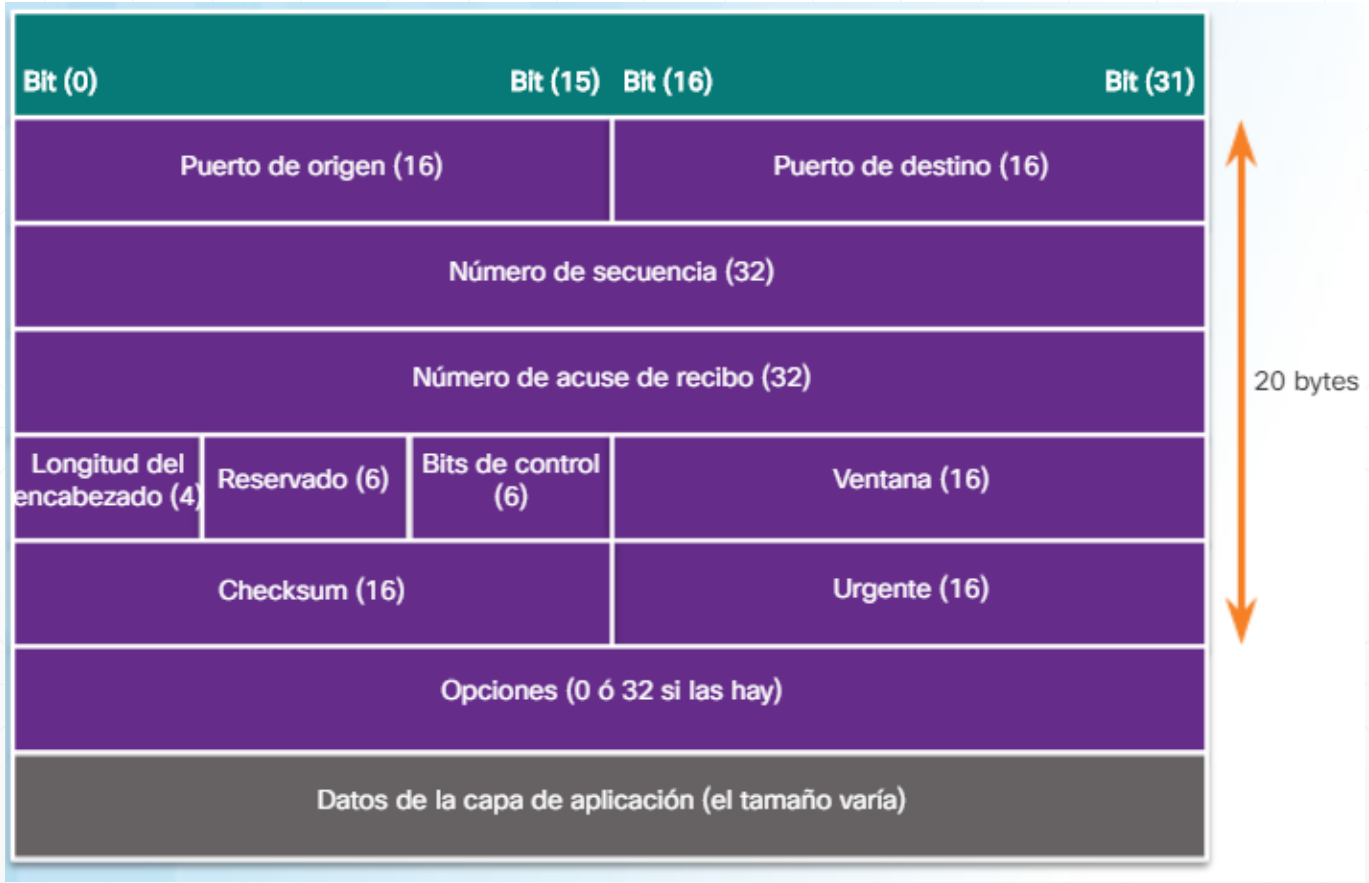
Características TCP

Control del flujo

- Los hosts de red tienen recursos limitados, como la memoria o la capacidad de procesamiento.
- Cuando TCP advierte que estos recursos están sobrecargados, puede solicitar que la aplicación emisora reduzca la velocidad del flujo de datos. Esto lo lleva a cabo TCP, que regula la cantidad de datos que transmite el origen.
- El control de flujo puede evitar la necesidad de retransmitir los datos cuando los recursos del host receptor están desbordados.

Ref: https://www.brianstorti.com/tcp-flow-control/

Segmento TCP

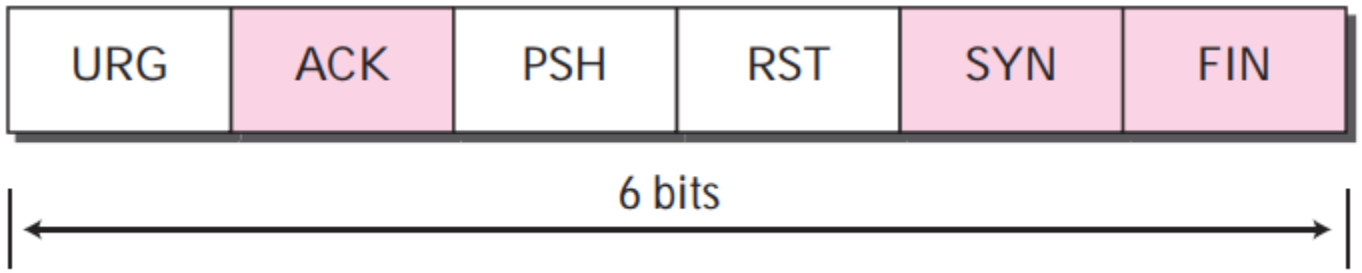


Encabezado TCP

- **Puerto de origen (16 bits) y puerto de destino (16 bits)** : se utilizan para identificar la aplicación.
- **Número de secuencia (32 bits)**: se utiliza para rearmar los datos.
- **Número de reconocimiento (32 bits)**: indica que los datos se han recibido y el siguiente byte esperado de la fuente.
- **Longitud del encabezado (4 bits)**: conocido como “desplazamiento de datos”. Indica la longitud del encabezado del segmento TCP.
- **Reservado (6 bits)**: este campo está reservado para el futuro.
- **Bits de control (6 bits)**: incluye códigos de bit, o marcadores, que indican el propósito y la función del segmento TCP.
- **Tamaño de la ventana (16 bits)**: indica la cantidad de bytes que se puedan aceptar por vez.
- **Checksum (16 bits)**: se utiliza para la verificación de errores en el encabezado y los datos del segmento.
- **Urgente (16 bits)**: indica si la información es urgente.

Bits de Control

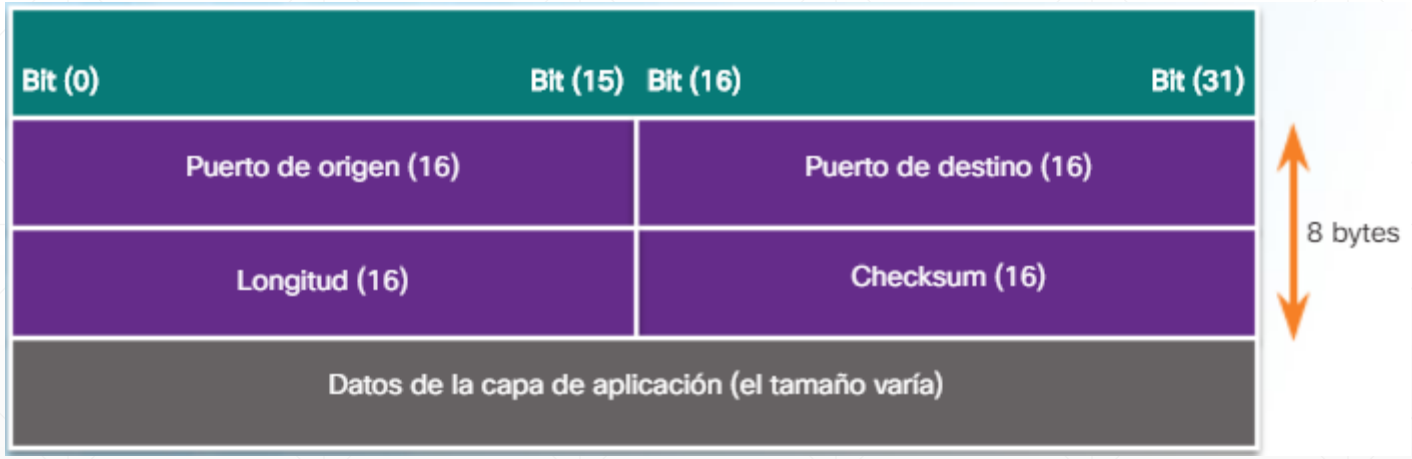
URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



Características UDP

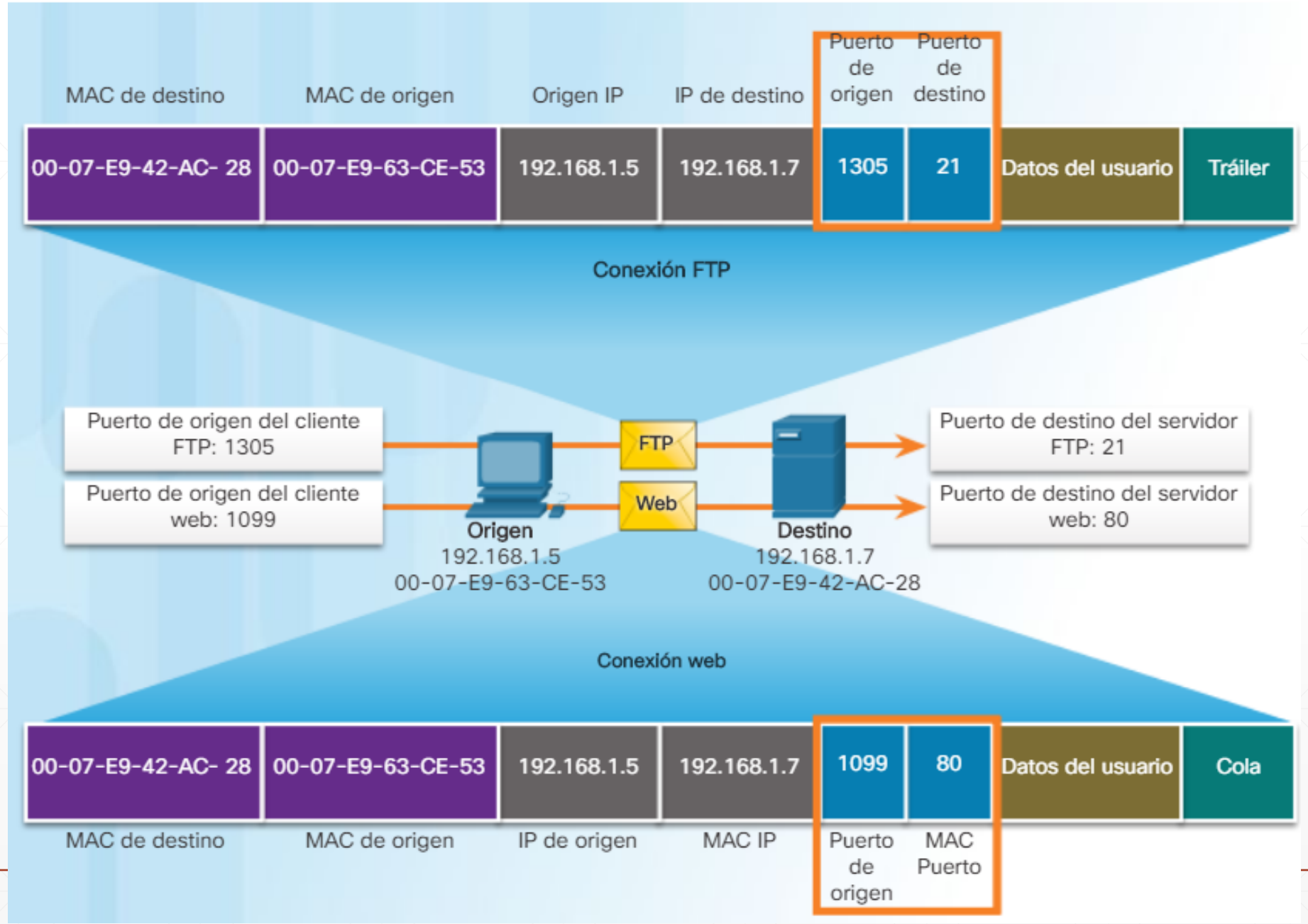
- Los datos se construyen en el orden en que se recibieron.
- Los segmentos perdidos no se vuelven a enviar.
- No hay establecimiento de sesión.
- No le informa al emisor sobre la disponibilidad de recursos.

Datagrama UDP



Pares de sockets

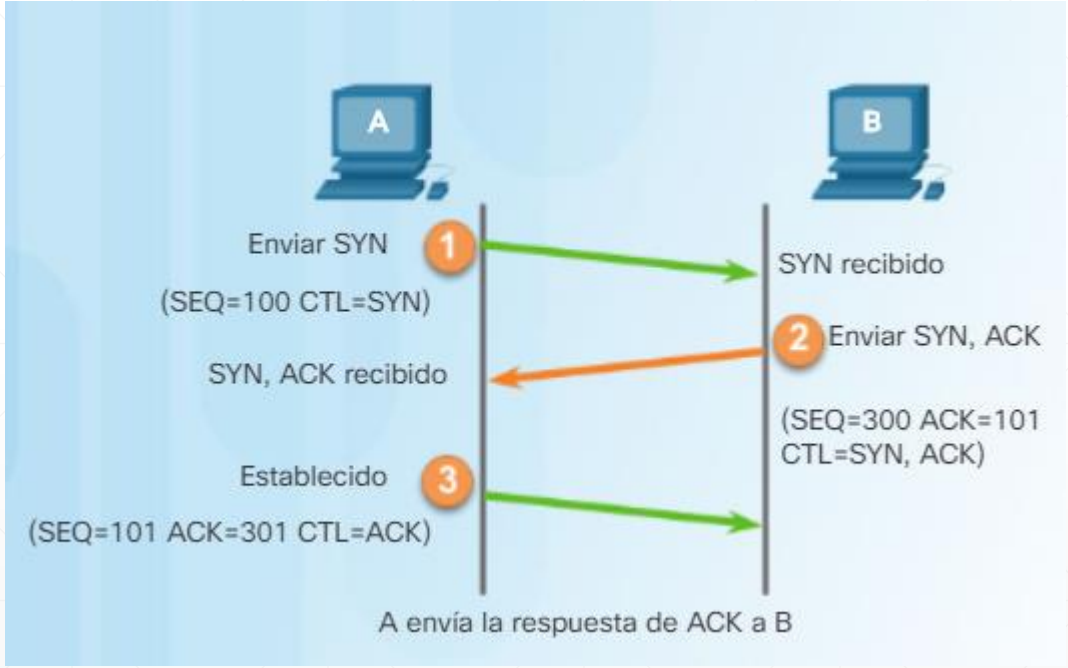
- Los puertos de origen y de destino se colocan dentro del segmento. Los segmentos se encapsulan dentro de un paquete IP. El paquete IP contiene la dirección IP de origen y de destino.
- Se conoce como socket a la combinación de la dirección IP de origen y el número de puerto de origen, o de la dirección IP de destino y el número de puerto de destino.
- El socket se utiliza para identificar el servidor y el servicio que solicita el cliente. Un socket de cliente puede ser parecido a esto, donde 1099 representa el número de puerto de origen: 192.168.1.5:1099
- El socket en un servidor web podría ser el siguiente: 192.168.1.7:80
- Juntos, estos dos sockets se combinan para formar un par de sockets: 192.168.1.5:1099, 192.168.1.7:80


MAC de destino
MAC de origen
Origen IP
IP de destino
Puerto de origen
Puerto de destino
00-07-E9-42-AC- 28
00-07-E9-63-CE-53
192.168.1.5
192.168.1.7
1305
21
Datos del usuario
Tráiler
Conexión FTP
Puerto de origen del cliente FTP: 1305
Puerto de origen del cliente web: 1099
FTP
Web
Origen
192.168.1.5
00-07-E9-63-CE-53
Destino
192.168.1.7
00-07-E9-42-AC-28
Puerto de destino del servidor FTP: 21
Puerto de destino del servidor web: 80
Conexión web
00-07-E9-42-AC- 28
00-07-E9-63-CE-53
192.168.1.5
192.168.1.7
1099
80
Datos del usuario
Cola
MAC de destino
MAC de origen
IP de origen
MAC IP
Puerto de origen
MAC Puerto

Establecimiento de conexiones (TCP three way handshake)

Una conexión TCP se establece en tres pasos:

- **Paso 1**: el cliente de origen solicita una sesión de comunicación de cliente a servidor con el servidor.
- **Paso 2**: el servidor reconoce la sesión de comunicación de cliente a servidor y solicita una sesión de comunicación de servidor a cliente.
- **Paso 3**: el cliente de origen reconoce la sesión de comunicación de servidor a cliente.

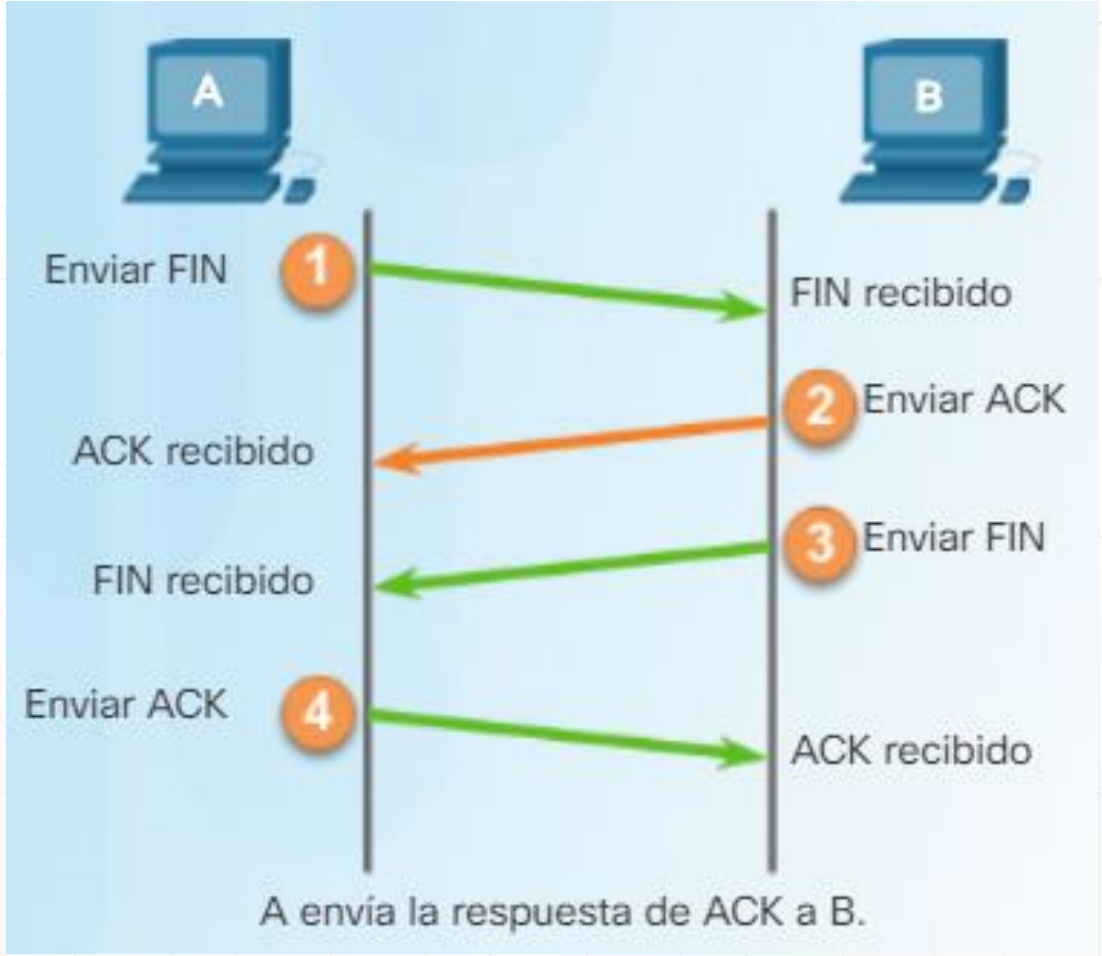


Finalización de la sesión TCP

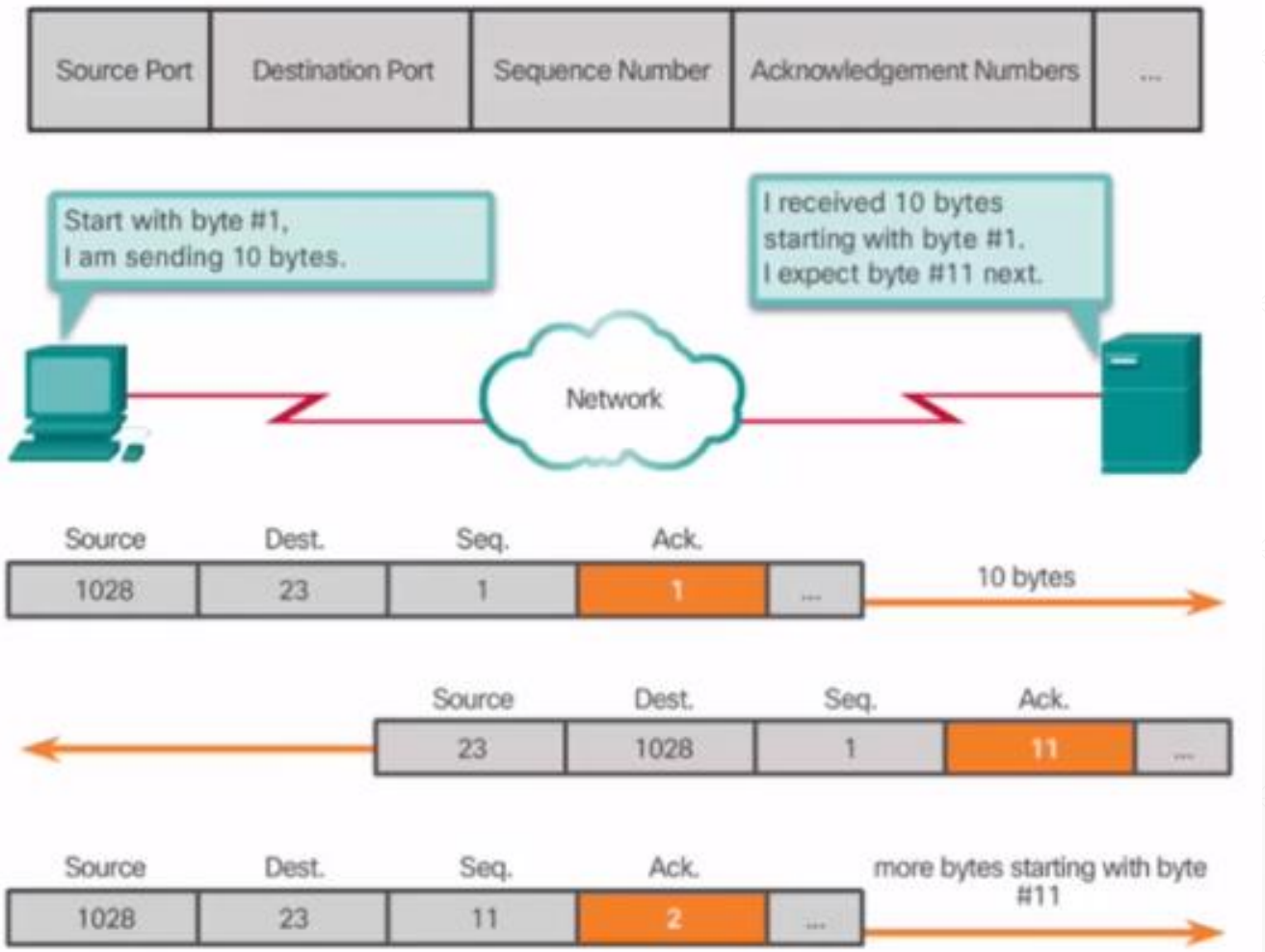
- Para cerrar una conexión, se debe establecer el marcador de control de finalización (FIN) en el encabezado del segmento.
- Para finalizar todas las sesiones TCP de una vía, se utiliza un enlace de dos vías, que consta de un segmento FIN y un segmento de reconocimiento (ACK).
- Por lo tanto, para terminar una conversación simple admitida por TCP, se requieren cuatro intercambios para finalizar ambas sesiones.

Finalización de la sesión TCP

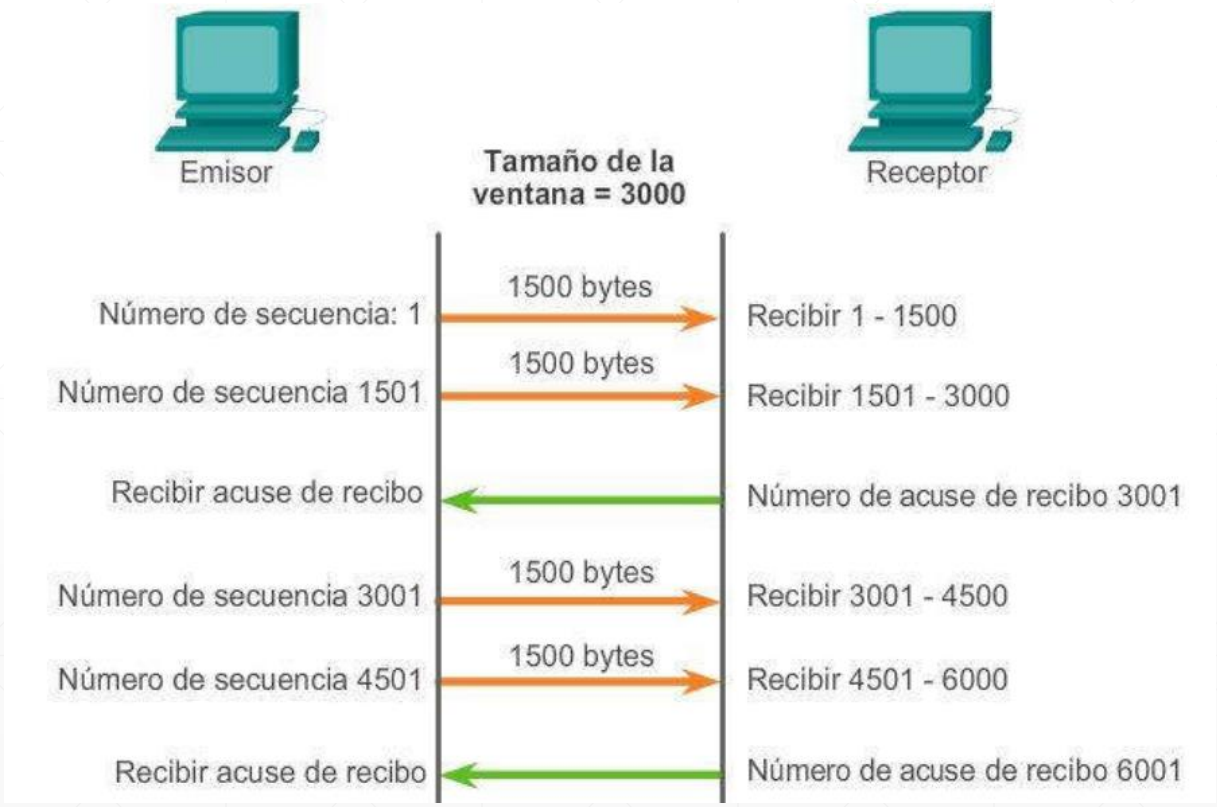
- **Paso 1**: cuando el cliente no tiene más datos para enviar en la transmisión, envía un segmento con el marcador FIN establecido.
- **Paso 2**: el servidor envía un ACK para reconocer el marcador FIN y terminar la sesión de cliente a servidor.
- **Paso 3**: el servidor envía un FIN al cliente para terminar la sesión de servidor a cliente.
- **Paso 4**: el cliente responde con un ACK para reconocer el recibo del FIN desde el servidor.



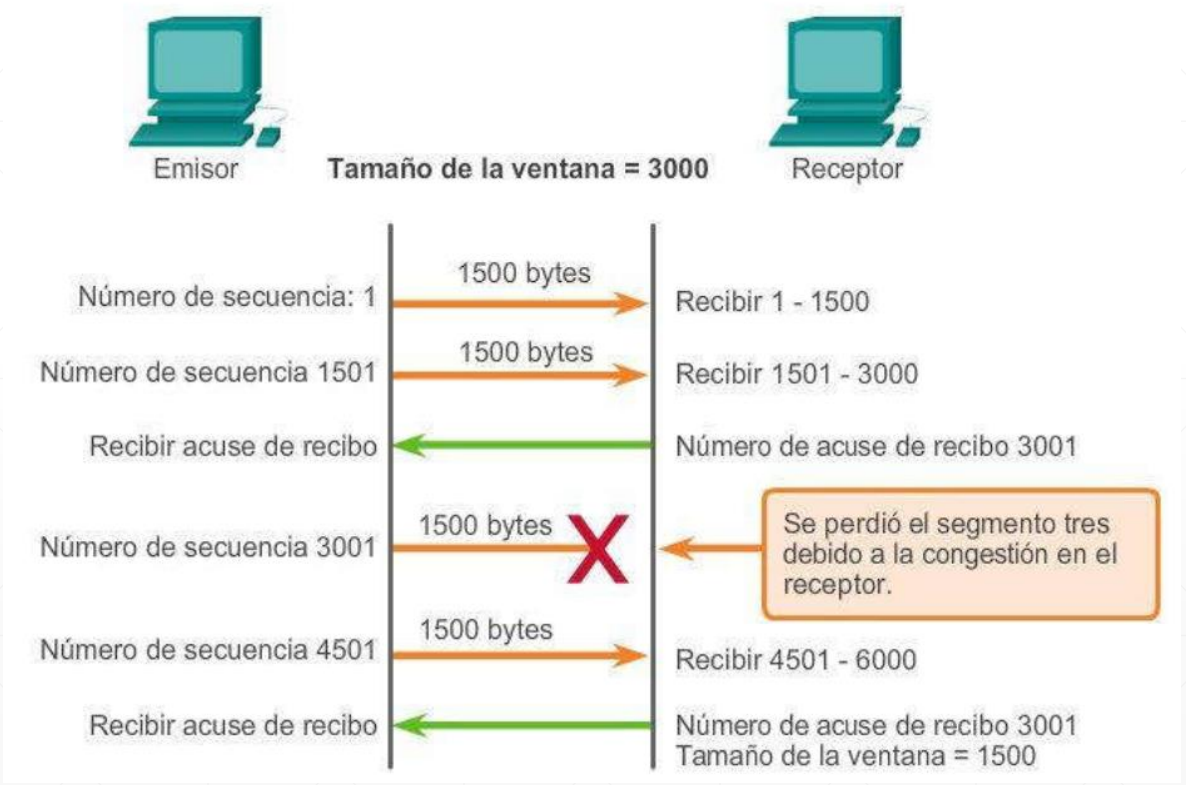
Window size


Source Port
Destination Port
Sequence Number
Acknowledgement Numbers
...
Start with byte #1,
I am sending 10 bytes.
I received 10 bytes
starting with byte #1.
I expect byte #11 next.
Network
Source
Dest.
Seq.
Ack.
1028
23
1
1
...
10 bytes
Source
Dest.
Seq.
Ack.
23
1028
1
11
...
Source
Dest.
Seq.
Ack.
1028
23
11
2
...
more bytes starting with byte
#11

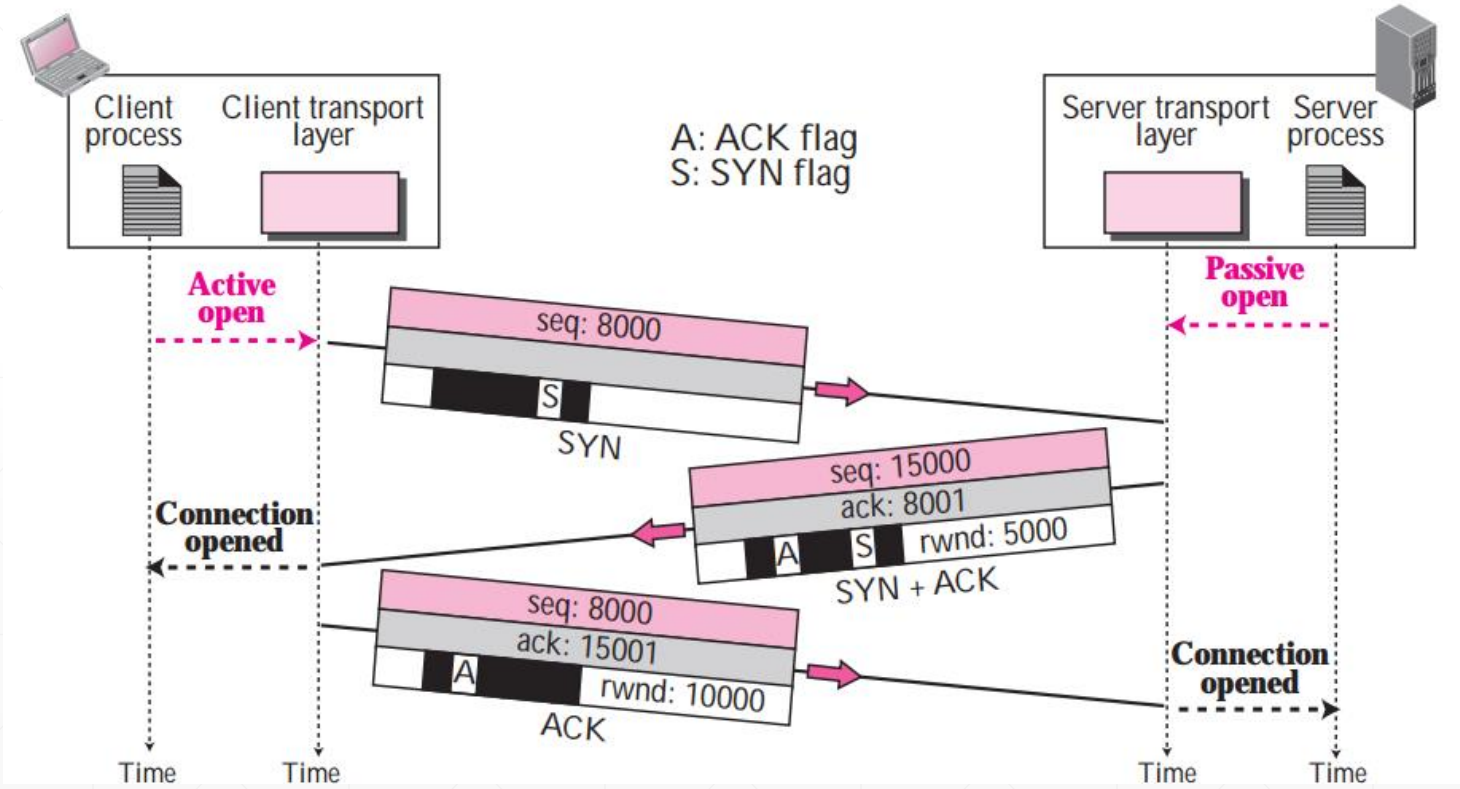
Acuse de recibido y tamaño de ventana TCP



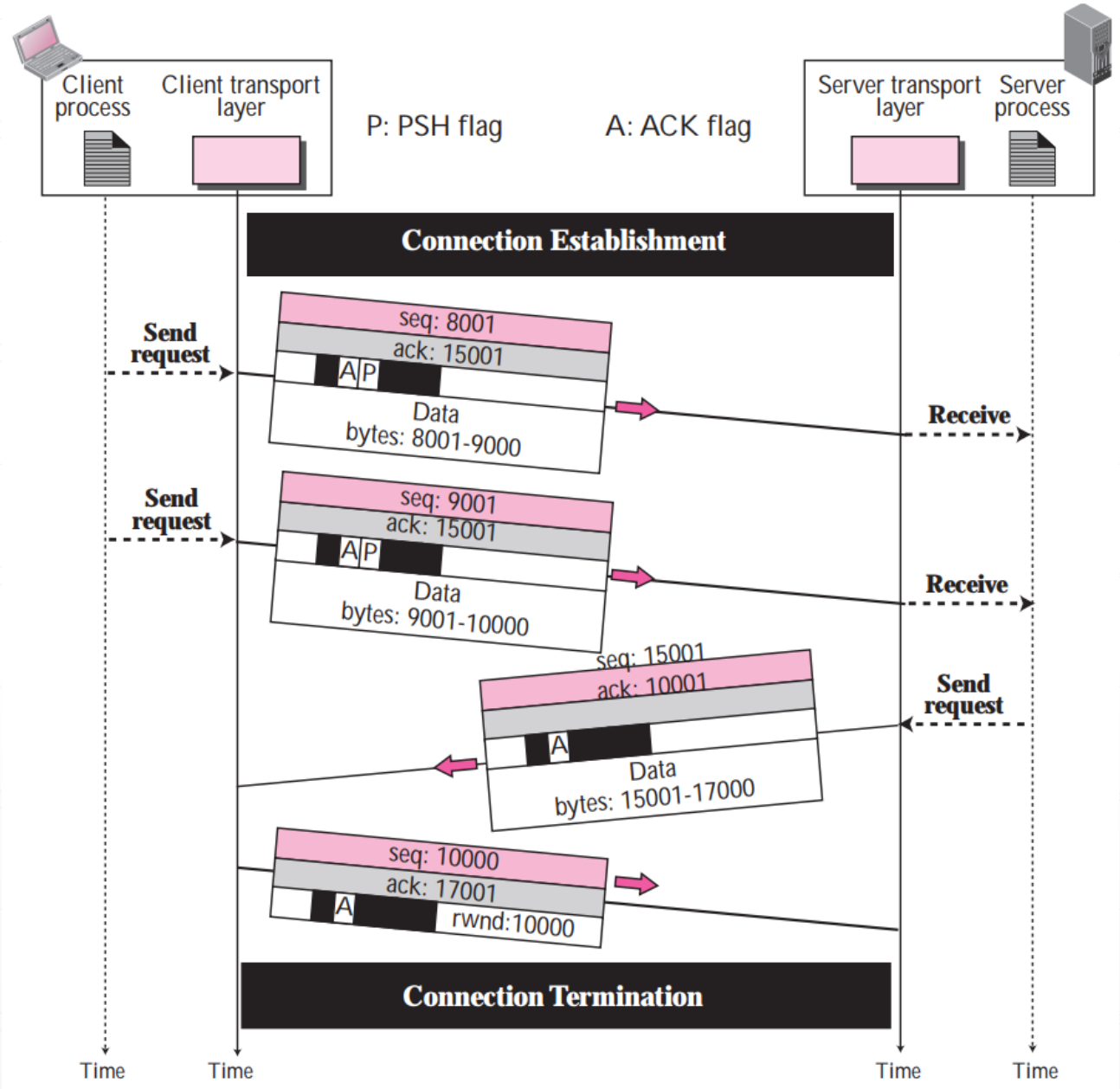
Congestión y control de flujo TCP



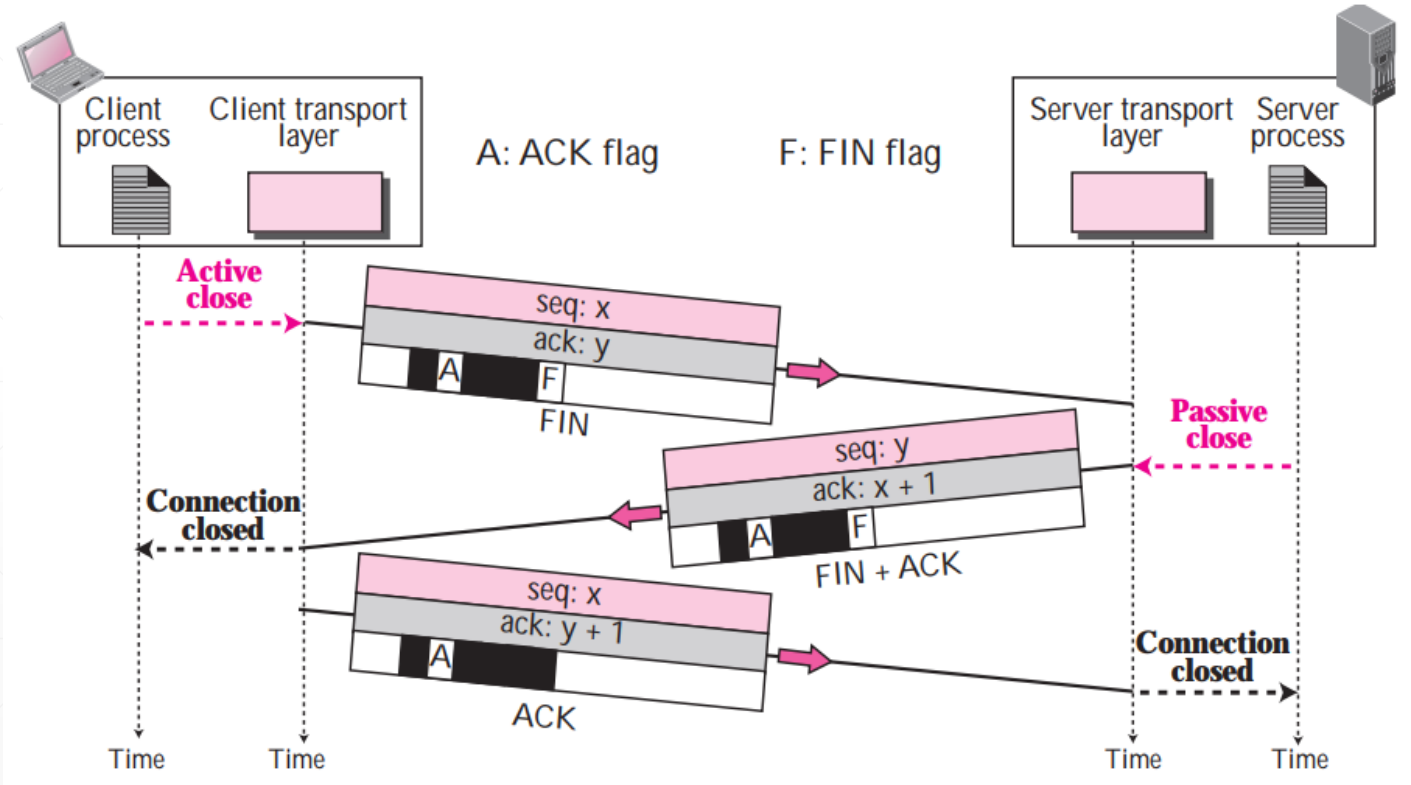
Connection stablishmet using three-way handshaking



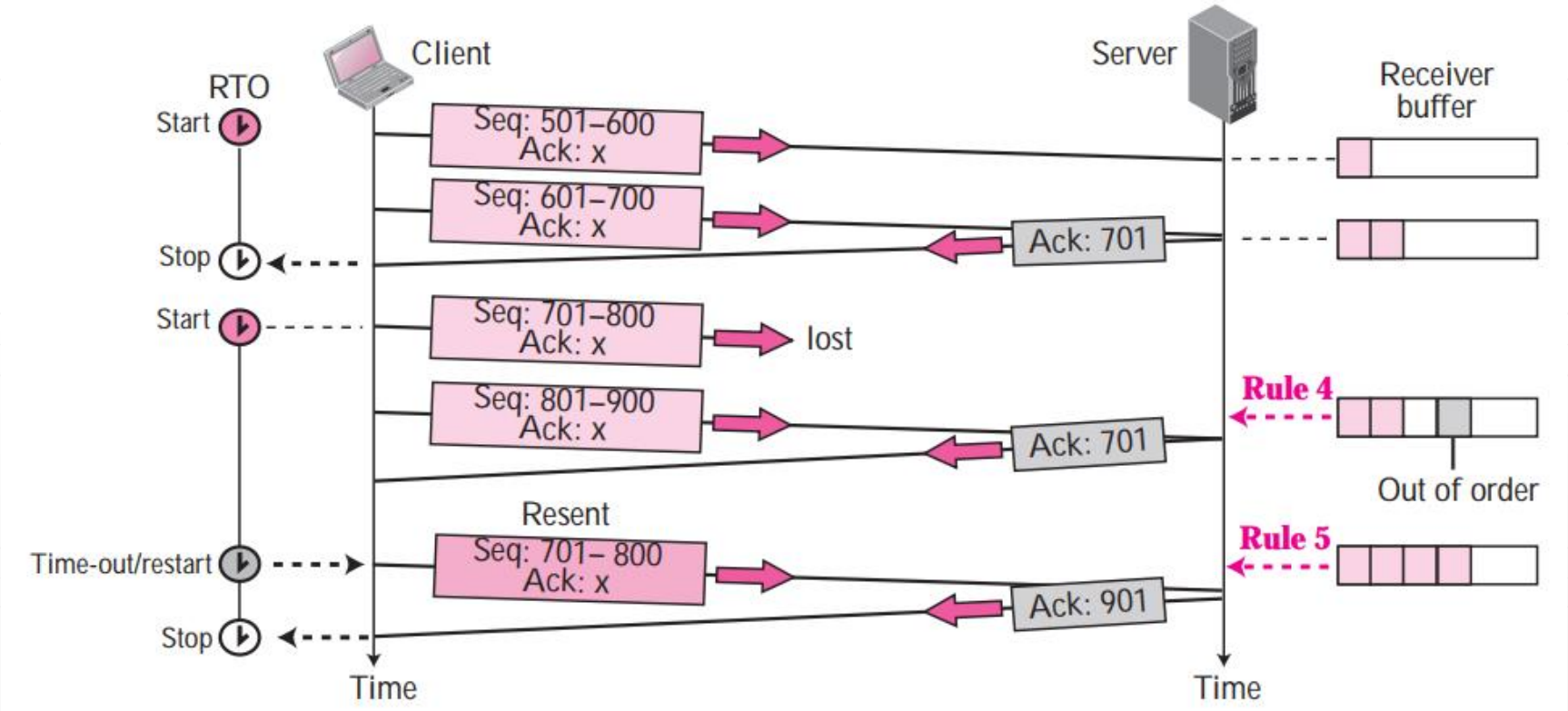
Data transfer



Connection termination using three-way handshaking



Lost packet


Client
Server
RTO
Start
Stop
Start
Time-out/restart
Stop
Seq: 501–600
Ack: x
Seq: 601–700
Ack: x
Ack: 701
Seq: 701–800
Ack: x
lost
Seq: 801–900
Ack: x
Ack: 701
Resent
Seq: 701– 800
Ack: x
Ack: 901
Receiver
buffer
Rule 4
Out of order
Rule 5
Time
Time