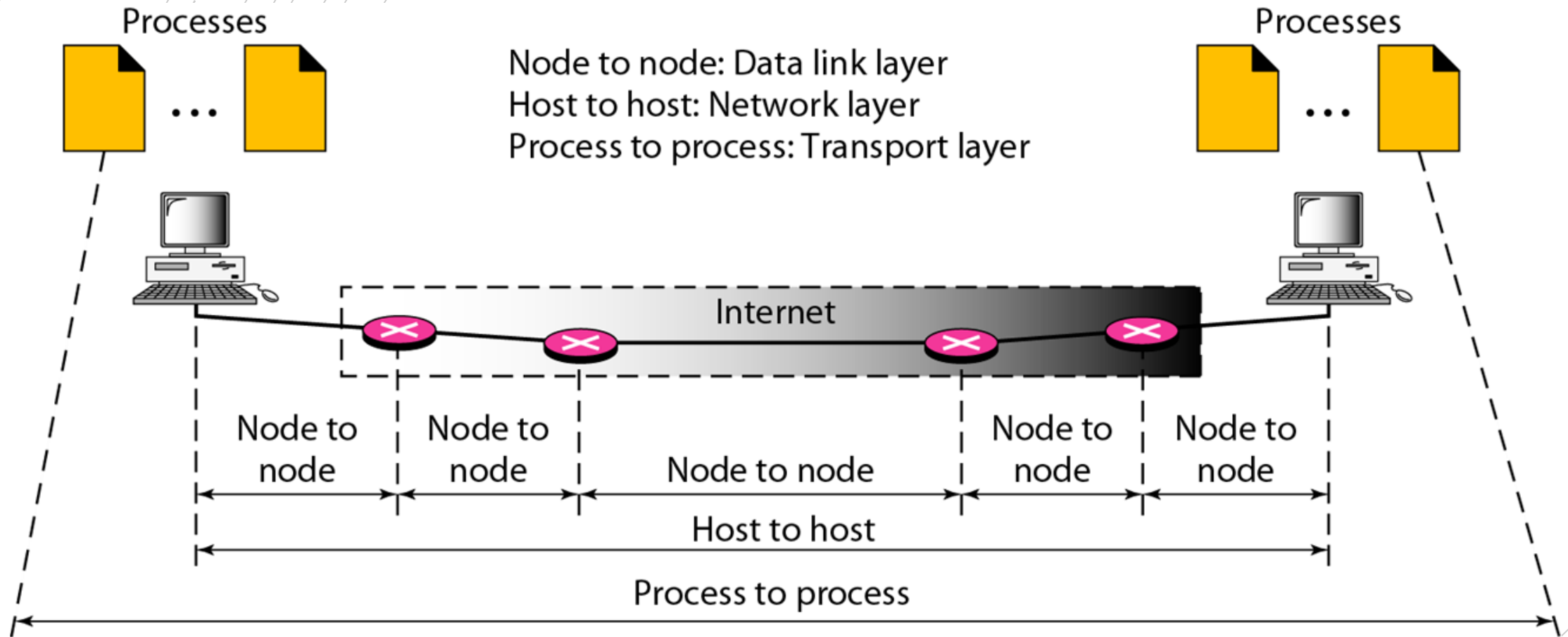
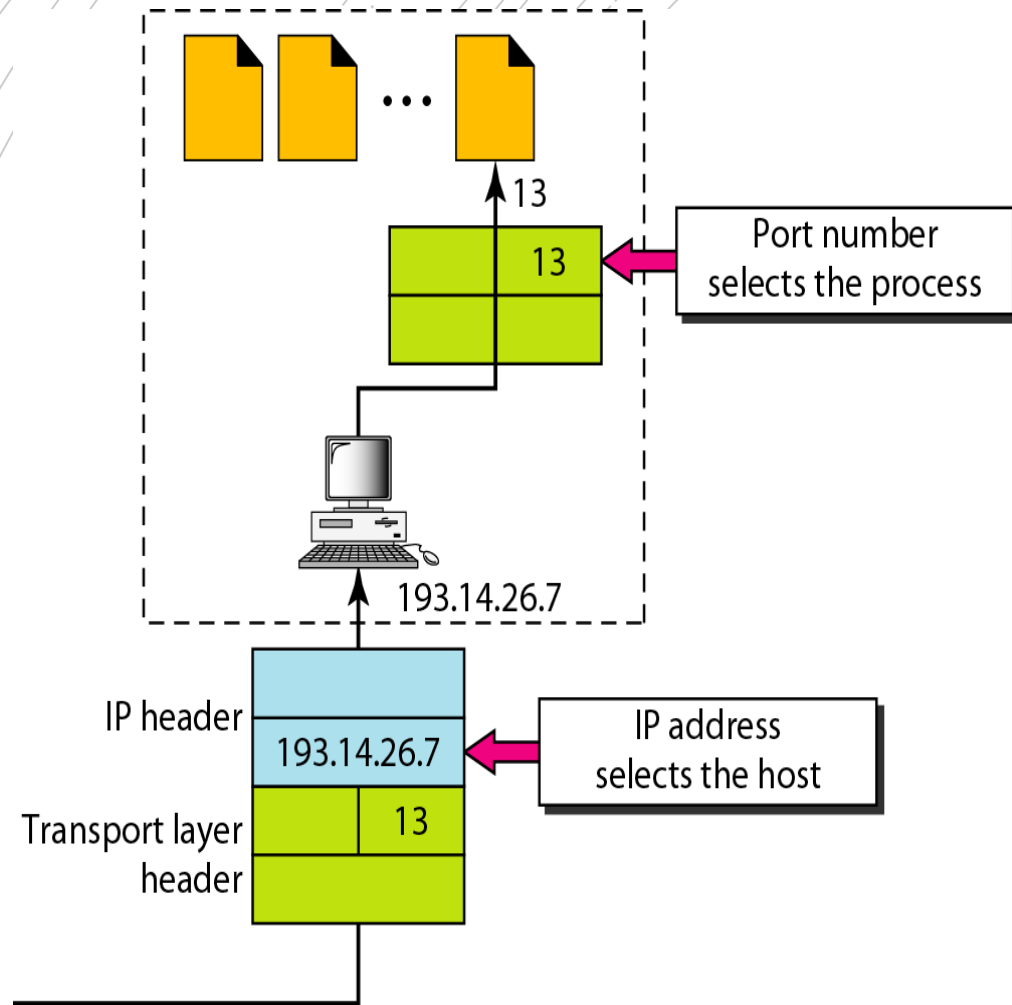


# LA CAPA TRANSPORTE

UDP-TCP

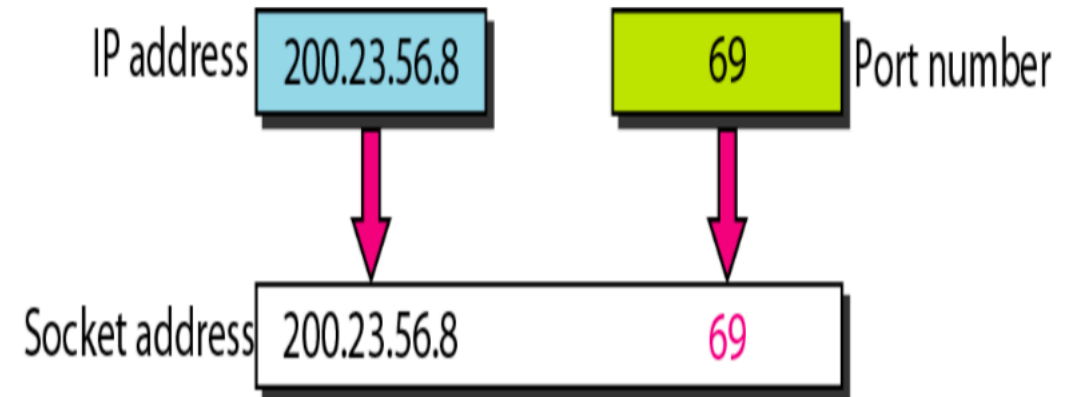
# Comparación de los tipos de entrega a nivel de las diferentes capas



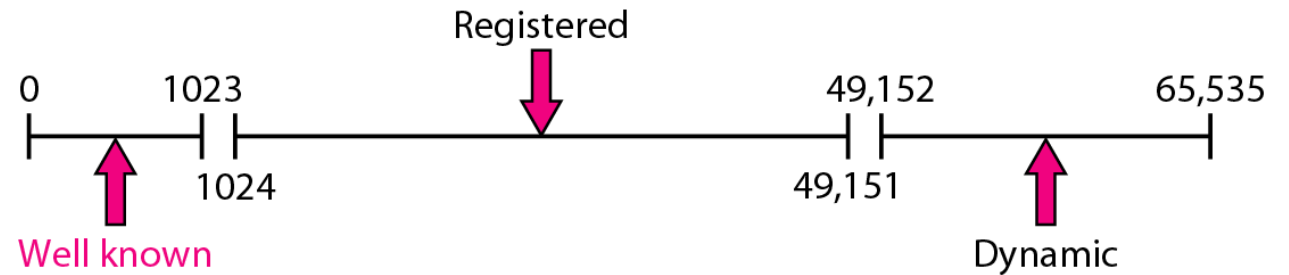


- La ip selecciona al Host
- El número de puerto selecciona el proceso

## Dirección de socket:

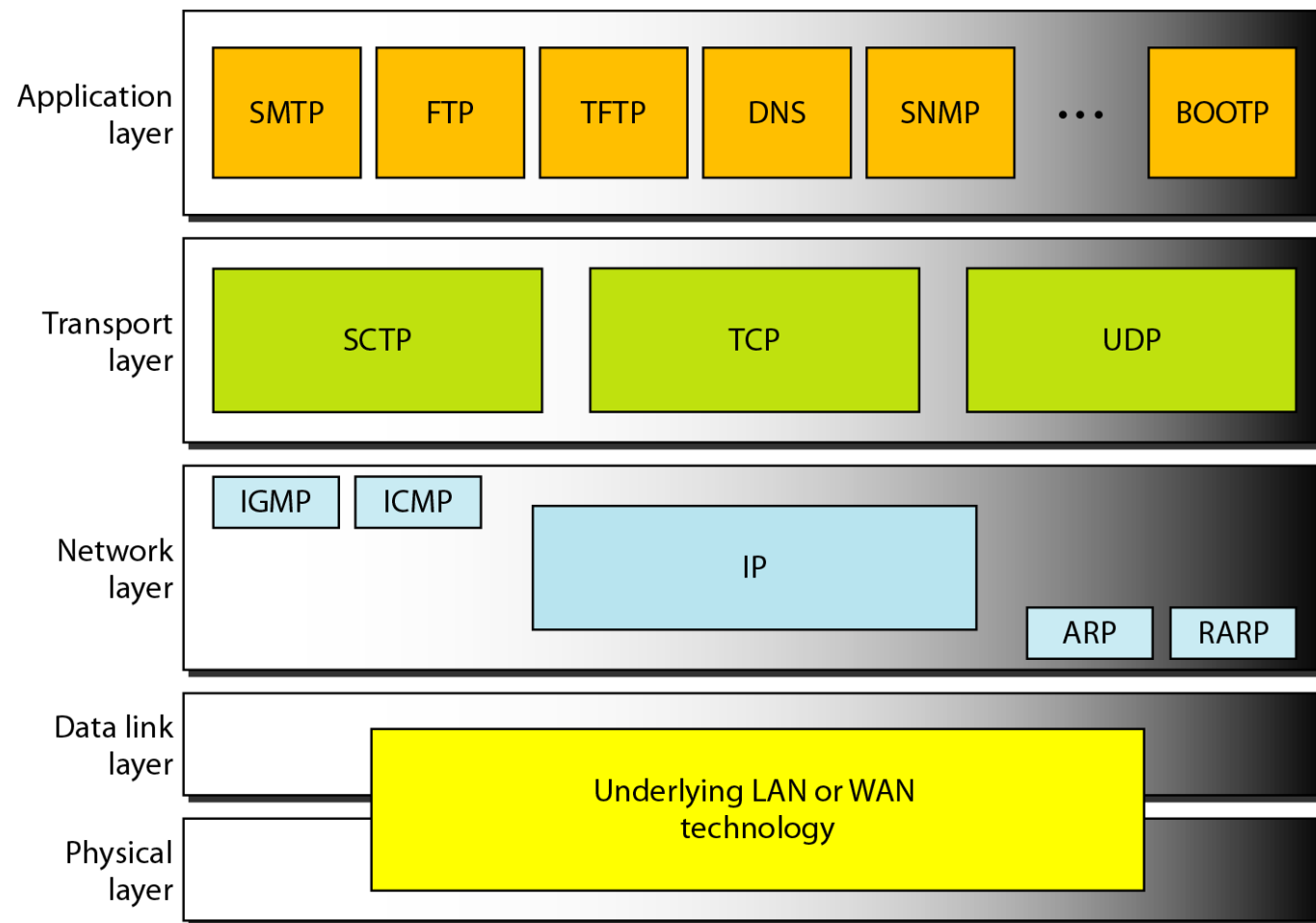


## LA IANA (Internet Assigned Numbers Authority)



- Los puertos “bien conocidos” (Well known) son utilizados para identificar servicios bien conocidos, como http (80) smtp (25) dns (53) ftp (21) etc.
- Los puertos “registrados”, pueden ser utilizados por diferentes servicios de comunicación en la red, se llaman así por que aplicaciones muy populares han registrado ese número de puerto (por ejemplo las bases de datos Oracle utilizan el puerto 1521)
- Los puertos “Dinámicos”, son asignados “aleatoriamente” por los sistemas operativos, para establecer comunicaciones regularmente hacia puertos “bien conocidos o Registrados

# Protocolos por capa



# PUERTOS BIEN CONOCIDOS EN SISTEMAS OPERATIVOS BASADOS EN UNIX

```
Manu — Manu@MacBook-Pro-de-Manuel — ~ — zsh — 80x24
Last login: Sun Mar 11 14:24:27 on console
[➔ ~ grep ftp /etc/services
ftp-data      20/udp      # File Transfer [Default Data]
ftp-data      20/tcp      # File Transfer [Default Data]
ftp           21/udp      # File Transfer [Control]
ftp           21/tcp      # File Transfer [Control]
ni-ftp        47/udp      # NI FTP
ni-ftp        47/tcp      # NI FTP
tftp          69/udp      # Trivial File Transfer
tftp          69/tcp      # Trivial File Transfer
sftp          115/udp     # Simple File Transfer Protocol
sftp          115/tcp     # Simple File Transfer Protocol
bftp          152/udp     # Background File Transfer Program
bftp          152/tcp     # Background File Transfer Program
softpc        215/udp     # Insignia Solutions
softpc        215/tcp     # Insignia Solutions
subntbcst_tftp 247/udp     # SUBNTBCST_TFTP
subntbcst_tftp 247/tcp     # SUBNTBCST_TFTP
mftp          349/udp     # mftp
mftp          349/tcp     # mftp
#             <veizades@ftp.com>
ftp-agent     574/udp     # FTP Software Agent System
ftp-agent     574/tcp     # FTP Software Agent System
#             Michael S. Greenberg <arnoff@ftp.com>
```

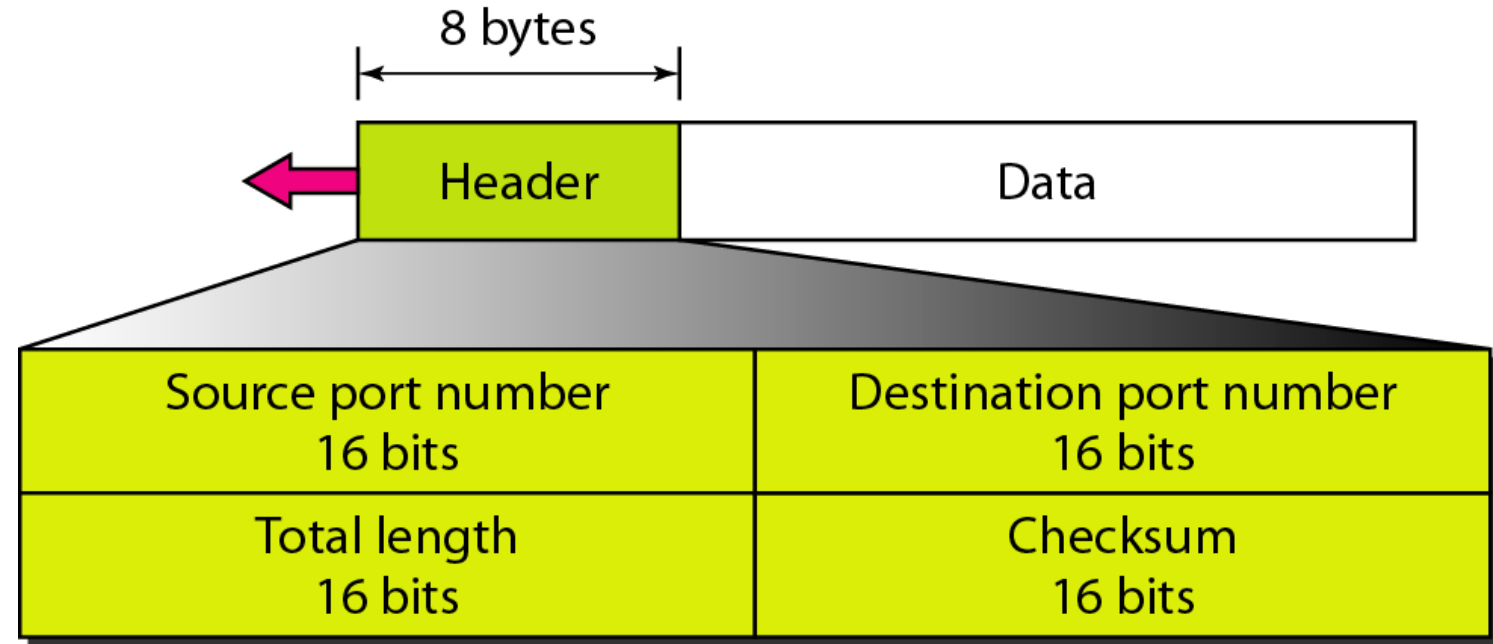
- Los puertos bien conocidos de los servicios que un host presta se pueden encontrar en `/etc/services`

## USER DATAGRAM PROTOCOL UDP

<i>Port</i>	<i>Protocol</i>	
7	Echo	Echoes a receive
9	Discard	Discards any dat
11	Users	Active users
13	Daytime	Returns the date
17	Quote	Returns a quote
19	Chargen	Returns a string
53	Nameserver	Domain Name S
67	BOOTPs	Server port to do
68	BOOTPc	Client port to do
69	TFTP	Trivial File Tran
111	RPC	Remote Procedu
123	NTP	Network Time P
161	SNMP	Simple Network
162	SNMP	Simple Network

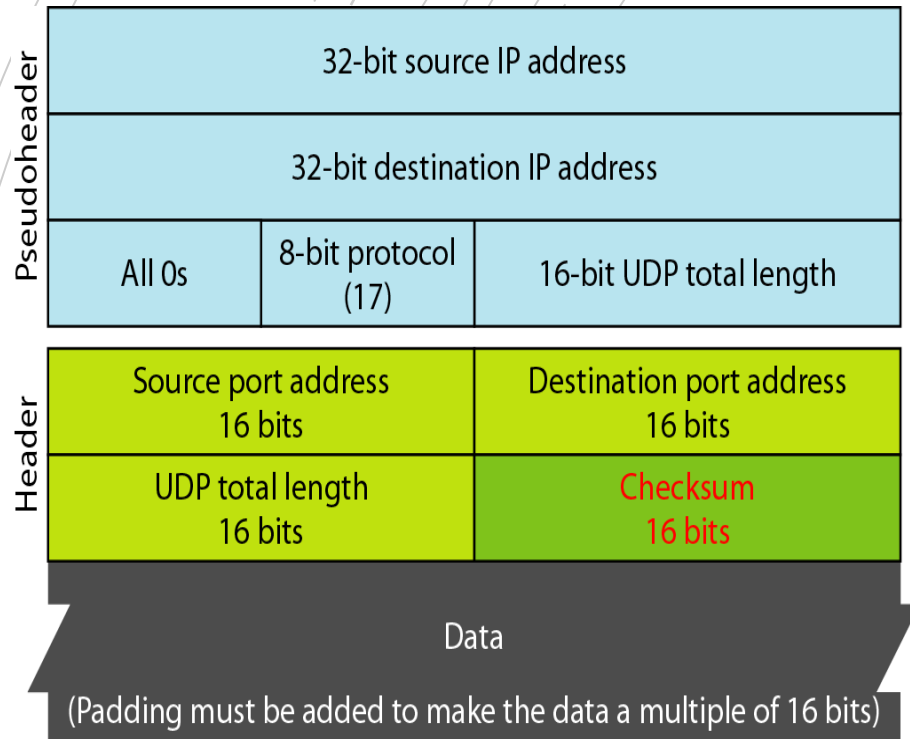
- El protocolo de Datagramas de Usuario (UDP), es un protocolo “No Orientado a la Conexión”
- No agrega ninguna información de control adicional a la de los servicios del protocolo de Internet (IP), exceptuando que provee comunicación “end to end” entre procesos, en lugar de host a host.

# FORMATO DE UDP

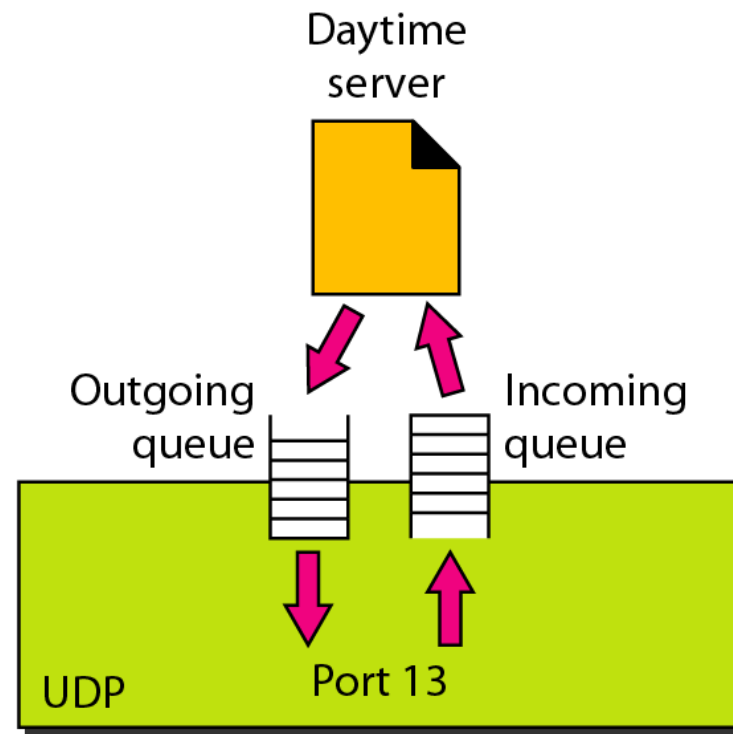
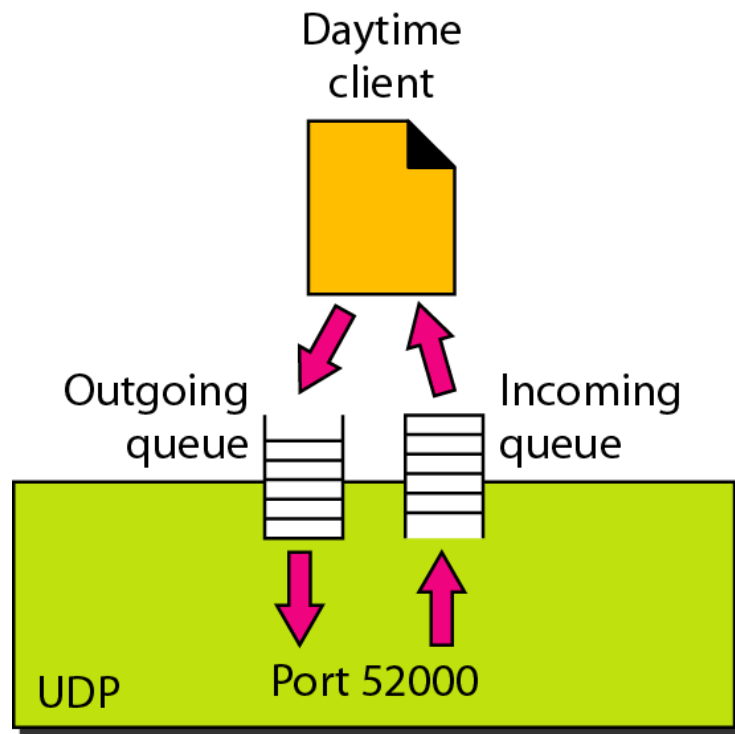




## DETECCION DE ERRORES EN UDP



- La detección de errores en UDP se realiza por medio del calculo de sumas de comprobación (Checksum)
- Cada uno de los elementos del Datagrama de usuario y un “PseudoEncabezado” se convierten a binario.
- Se dividen en grupos de 2bytes (palabras de 16 bits)
- Los datos se rellenan al final con 0 (padding) para hacer que los datos sean múltiplos de 16



## MANEJO DE COLAS DE UDP

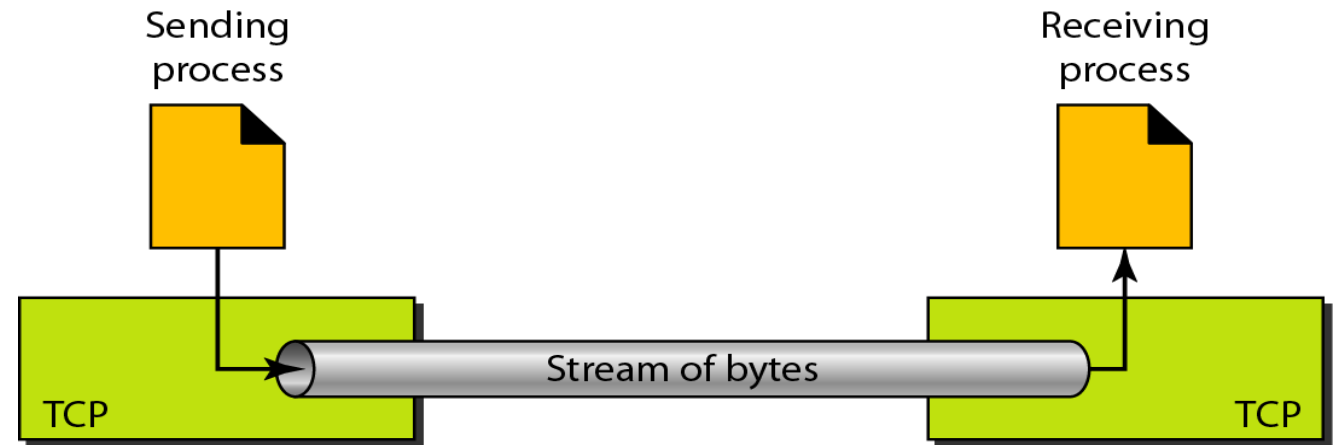
<i>Port</i>	<i>Protocol</i>	<i>Description</i>
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

## TRANSMISSION CONTROL PROTOCOL - TCP

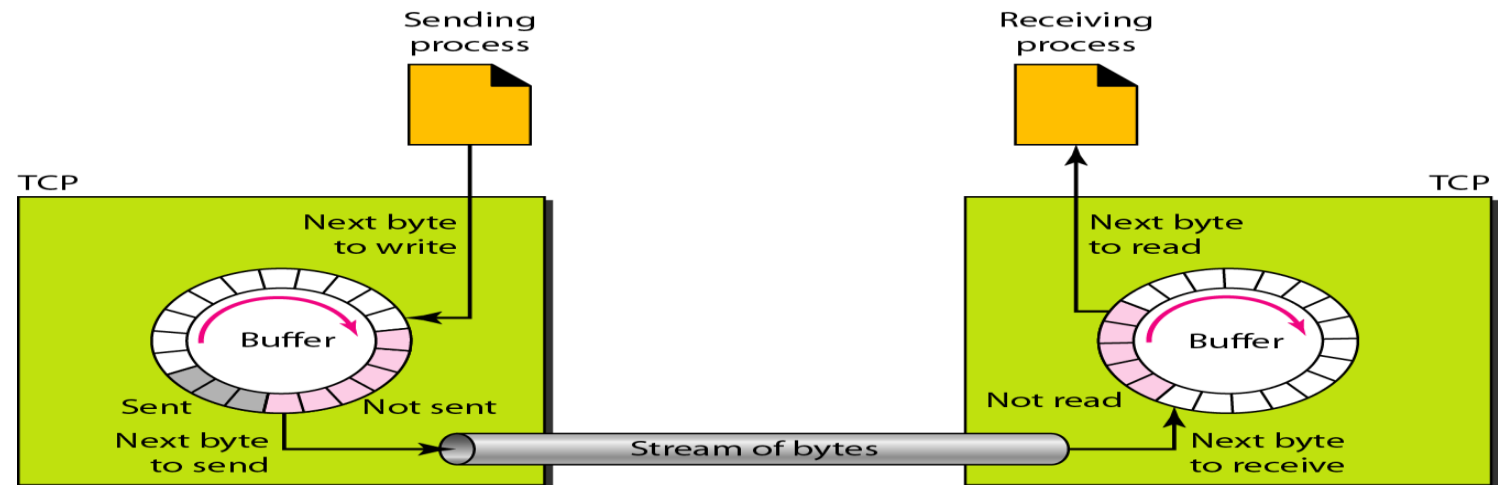
- TCP es un protocolo “Orientado a la conexión”
- Crea una conexión virtual entre dos hosts que utilizan
- TCP, para enviar datos entre si.
- TCP utiliza mecanismos de control de flujo y errores a nivel de la capa de transporte

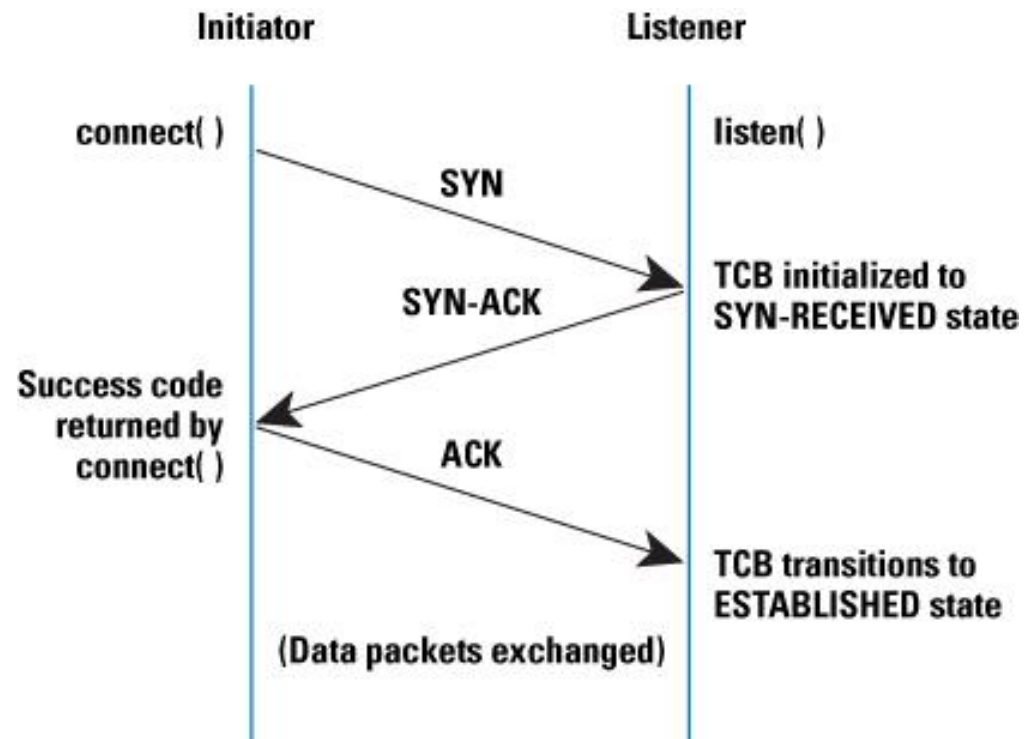
# Transmisión en TCP

EN TCP, LA TRANSMISIÓN ES UNA “CORRIENTE” DE BYTES



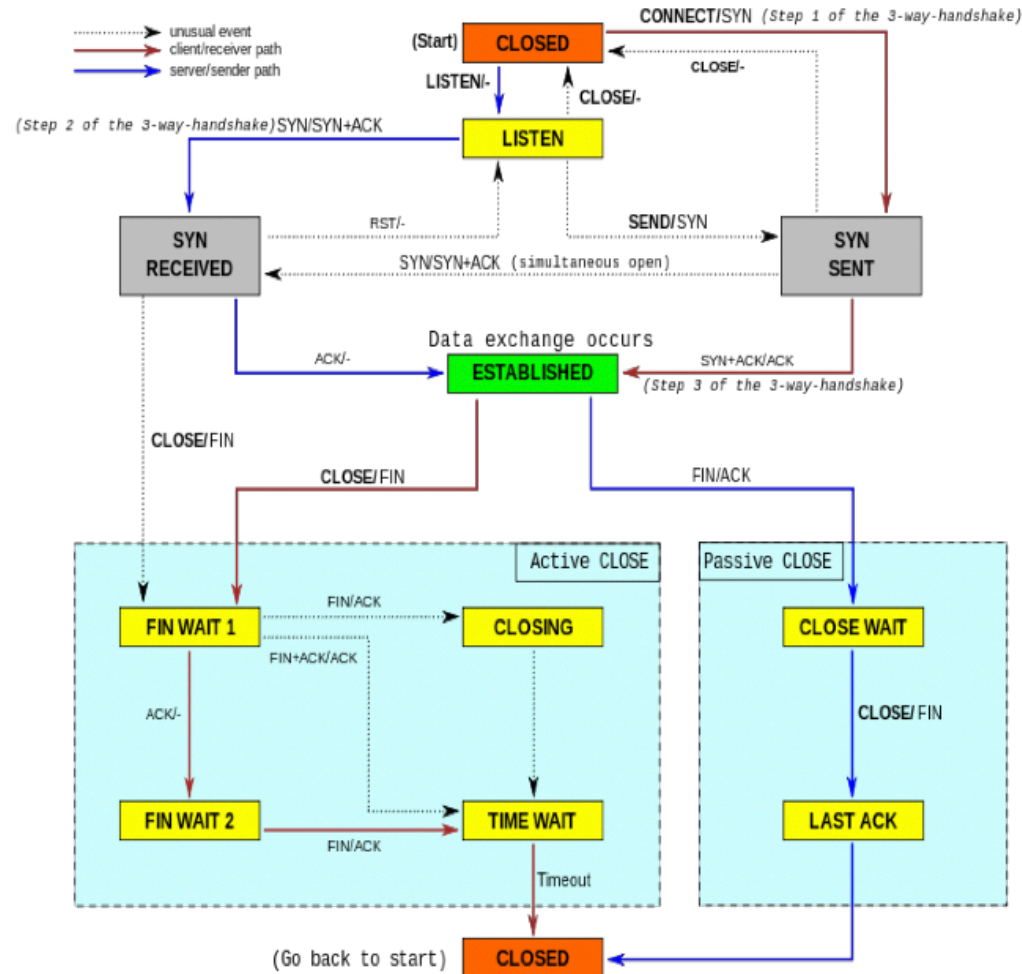
PARA CONTROLAR LA CORRIENTE DE BYTES SE UTILIZAN BUFFERS PARA MANEJAR LAS COLAS DE RECEPCIÓN Y ENVÍO



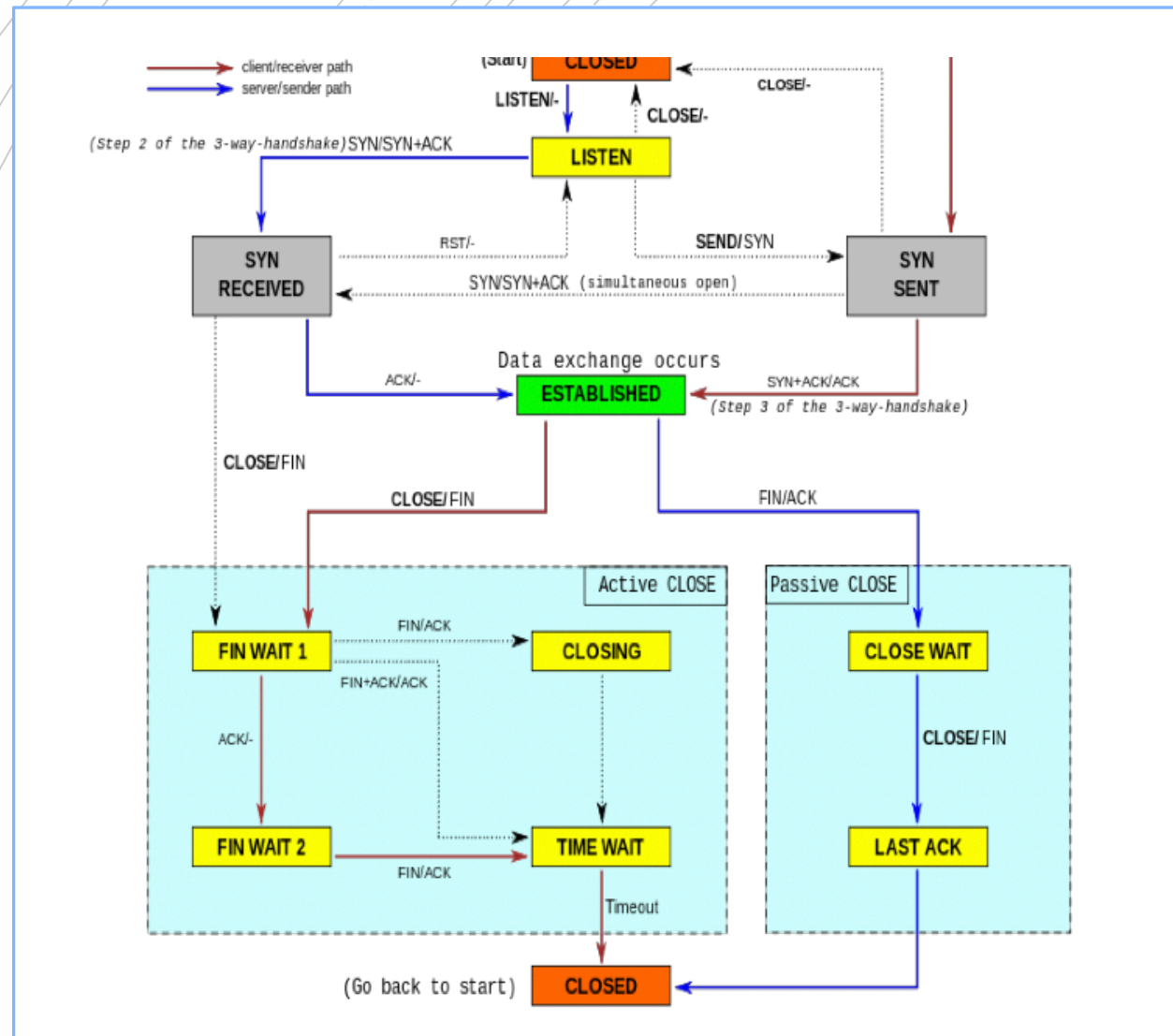


## Three-way handshake

# Estado de una conexión TCP



- **LISTEN(servidor):** la espera por una conexión remota
- **SYN-SENT (cliente):** la espera después de enviar una solicitud de conexión
- **SYN-RECEIVED (servidor):** se espera por la confirmación (Ack), después de ambos recibieron la solicitud de conexión.
- **ESTABLISHED (ambos):** una conexión abierta, se envían datos
- **FIN-WAIT-1 (ambos):** espera por un request de finalización de la conexión o por un ACK de un request de finalización previo, de una conexión TCP remota
- **FIN WAIT-2 (ambos):** representa la espera de un request de finalización, de una conexión TCP remota



- **CLOSE-WAIT(ambos):** espera por un requerimiento de finalización de la conexión local
- **CLOSING(ambos):** espera por un ACK de la solicitud de finalización de la conexión desde una conexión TCP remota
- **LAST-ACK (ambos):** espera por un ACK de un request de finalización previamente enviado a la conexión TCP REMOTA
- **TIME WAIT(cualquiera):** timeout máximo para esperar que el TCP remoto recibió el ACK de finalización (este estado puede durar hasta 4 minutos)
- **CLOSED (ambos):** Indica que no hay conexiones

# Secuencias

- Los bytes de datos que se transfieren en cada conexión son numerados por TCP (secuencias).
- Esta numeración inicia con un número generado aleatoriamente
- El valor en el campo de secuencia del segmento TCP, define el número del primer byte de datos contenido en ese segmento
- El valor del campo de acuse (acknowledgment) en el segmento, define el número del siguiente byte que el receptor espera. Este valor es acumulativo.



# Banderas de control

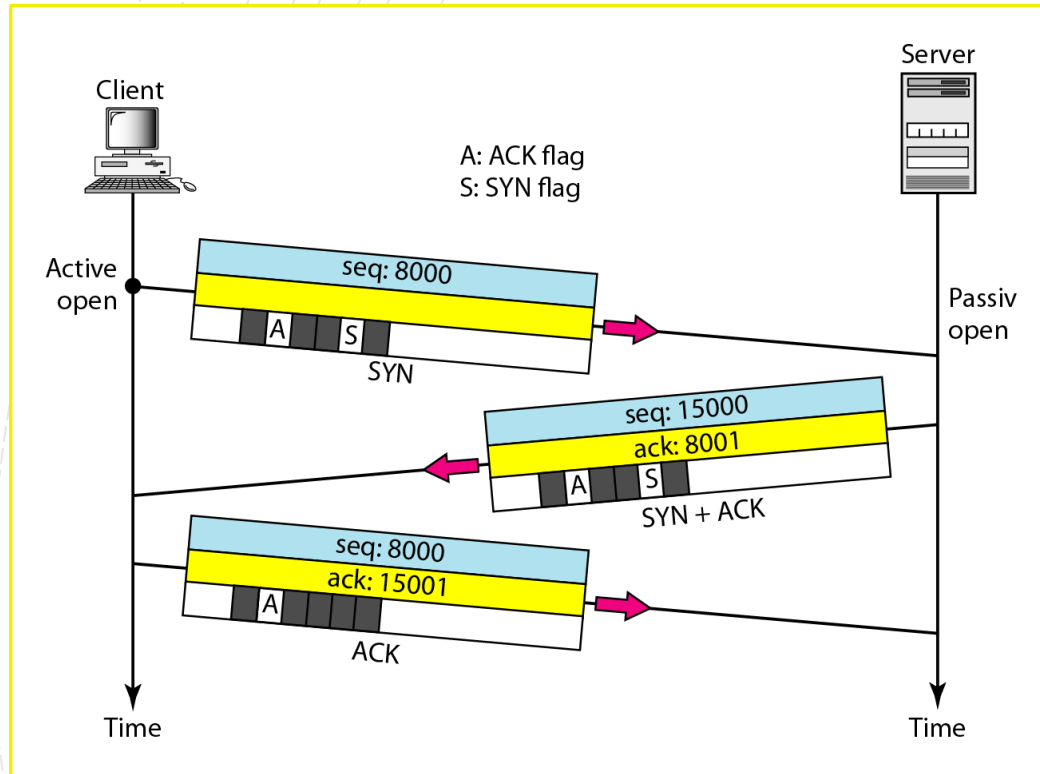
- **URG:** El puntero urgente es válido
- **ACK:** Acuse válido
- **PHS:** Requerimiento de Push (Envío de datos)
- **RST:** Reinicio de Conexión
- **SYN:** Sincronización de números de secuencia
- **FIN:** Terminar la conexión

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

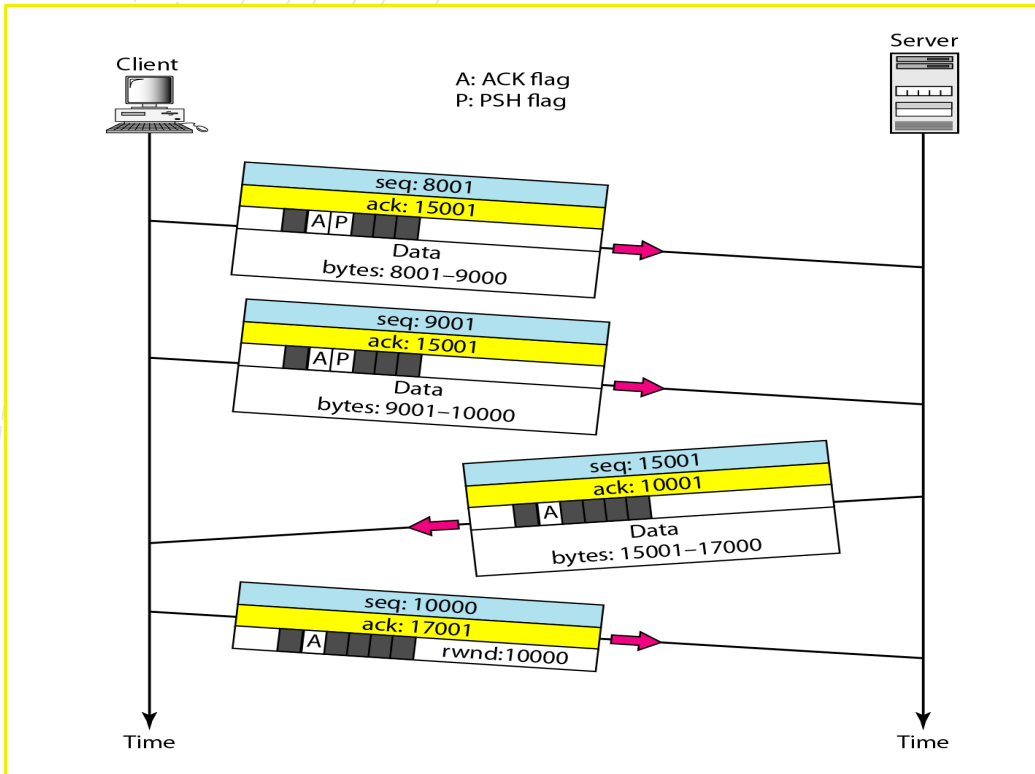


# Conexión utilizando un protocolo de tres vías



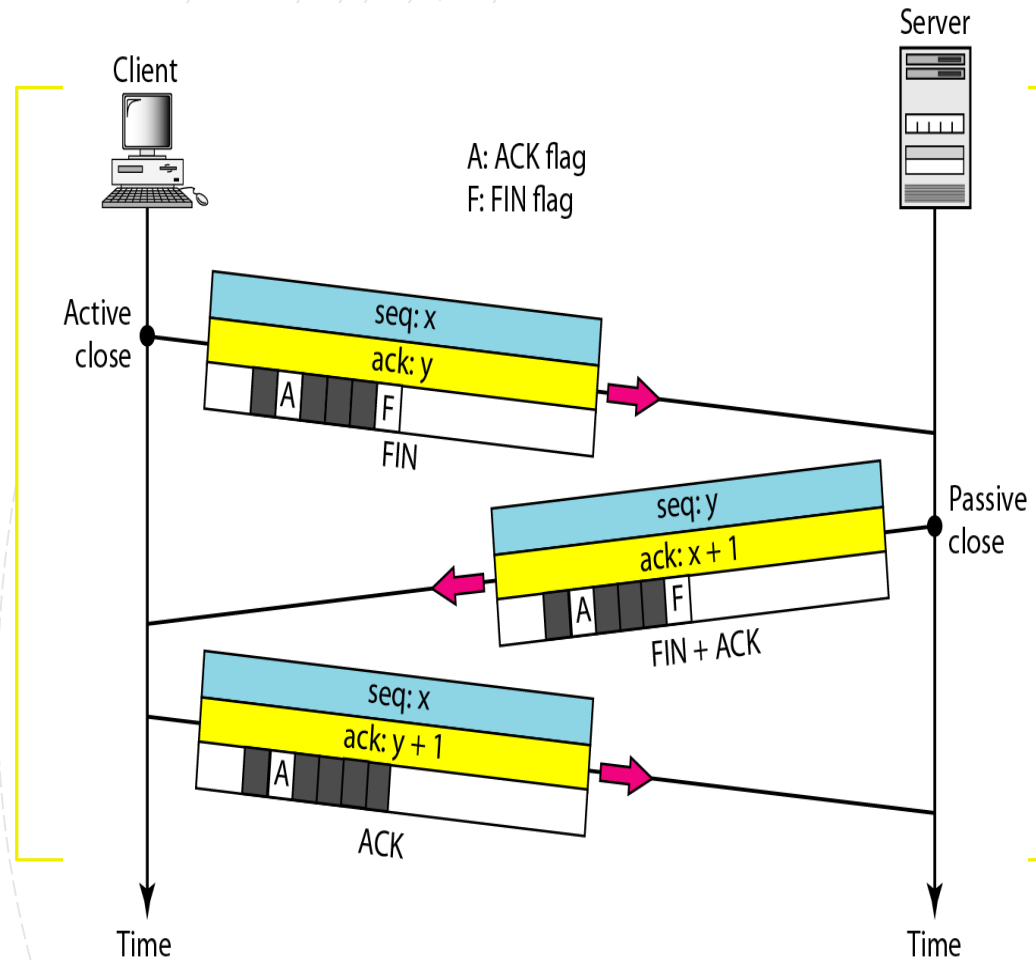
- Un segmento “SYN”, no lleva datos, pero consume secuencias
- Un segmento “SYN-ACK”, no lleva datos, pero consume números de secuencia
- Un segmento “ACK”, no consume un número de secuencia, siempre y cuando no lleve datos (piggyback)
- El servidor (quien provee el servicio), espera la conexión “Pasivamente”
- El cliente, que utilizara el servicio, solicita la conexión “Activamente”

# Transferencia de datos



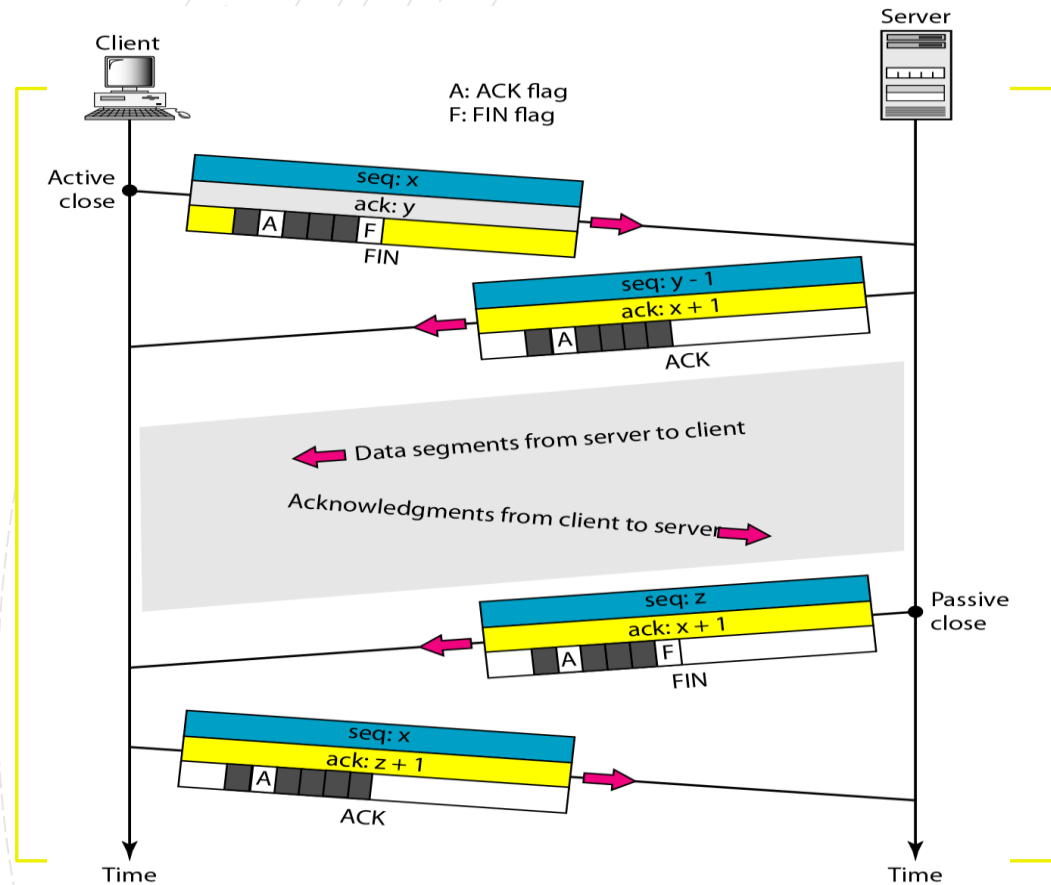
- Luego de la negociación o “handshake” de la conexión es posible el envío de segmentos de datos
- La bandera “PUSH” se activa en conjunto con “ACK”

# Finalización de la conexión



- Para terminar una conexión, uno de los pares que están comunicándose debe iniciar la solicitud de desconexión "Activamente", ajustando la bandera "FIN" en conjunto con "ACK"
- Quien recibe la solicitud de desconexión la realiza de forma pasiva y envía un acuse de recibido de esta solicitud, enviando una solicitud de desconexión también (FIN-ACK), incrementando la secuencia.
- FIN, consume una secuencia siempre y cuando lleve datos. FIN-ACK consume siempre un número de secuencia

# Finalización de la conexión

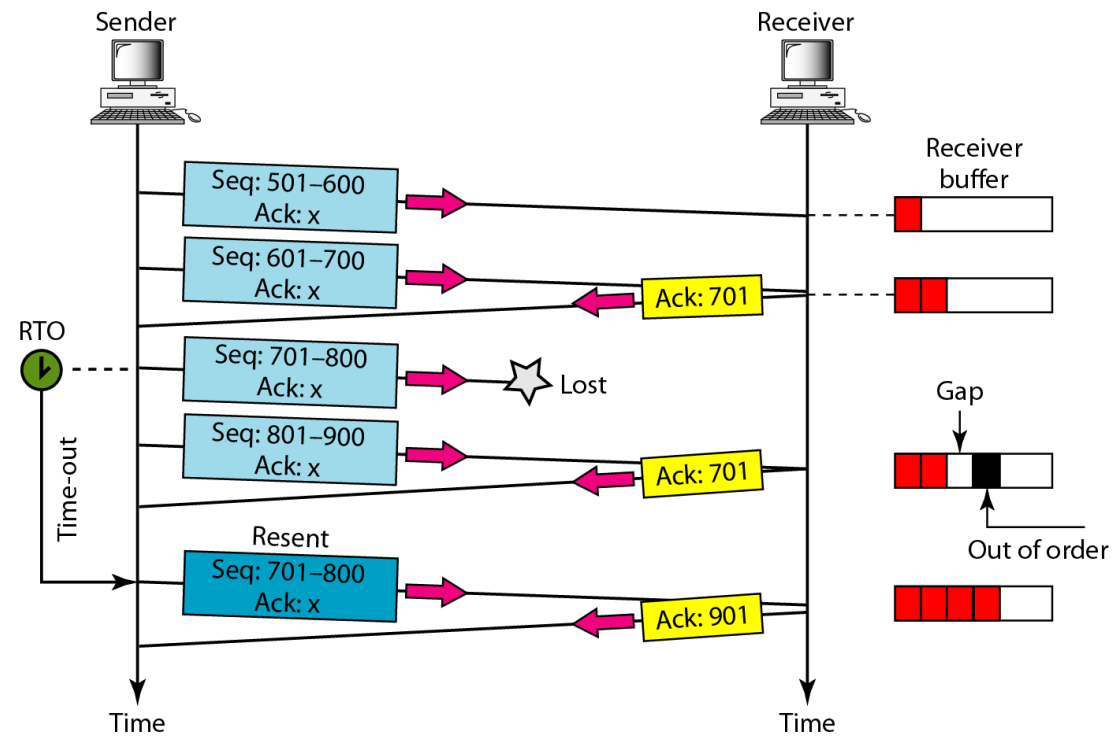


- El cliente no tiene más datos por enviar, entonces solicita activamente la desconexión
- El servidor acusa de recibida la solicitud de desconexión, pero el aún no la cierra hasta que concluya su transmisión de datos (no activa la bandera FIN)
- El cliente al no recibir la solicitud de desconexión del servidor, se queda con la conexión en modo "Half Close"
- Al terminar la transferencia de datos el servidor envía la solicitud de desconexión y el cliente acusa de recibido para completar el cierre de conexión

A large red speech bubble with a white outline, pointing downwards. Inside the bubble, the word '¡IMPORTANTE!' is written in white, bold, uppercase letters.

**¡IMPORTANTE!**

- Los segmentos ACK, no consumen números de secuencia
- Los ACK no tienen timer de retransmisión
- Los datos pueden arribar fuera de orden y son almacenados temporalmente por el receptor
- TCP garantiza que los segmentos fuera de orden sean ordenados previo a su entrega a la capa superior



## Perdida de Segmentos y Reenvío