



ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS
Área de Ingeniería Telemática

TCP: Características

Establecimiento y finalización de conexiones

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Arquitectura de Redes, Sistemas y Servicios
3º Ingeniería de Telecomunicación



Temario

1. Introducción
2. Protocolos y arquitectura
3. Redes de área local
4. Protocolos de Internet
5. Conmutación de circuitos
6. Conmutación de paquetes
7. Gestión de recursos en conmutadores
8. Protocolos de control de acceso al medio



Temario

1. Introducción
2. Protocolos y arquitectura
3. Redes de área local
4. Protocolos de Internet
 - Nivel de red
 - Nivel de transporte
 - UDP
 - **TCP**
 - Servicios
5. Conmutación de circuitos
6. Conmutación de paquetes
7. Gestión de recursos en conmutadores
8. Protocolos de control de acceso al medio



Objetivos

- Ver las características de TCP
- Especialmente qué quiere decir que sea “orientado a conexión”



Contenido

- Introducción
- Demultiplexación en TCP
- Formato del segmento TCP
- Gestión de conexiones



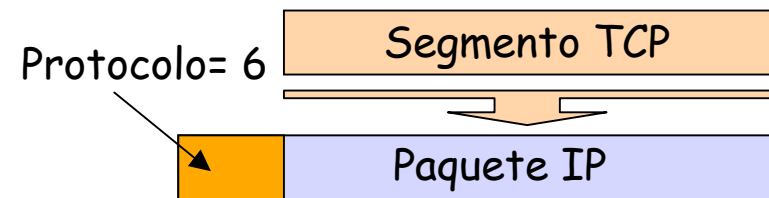
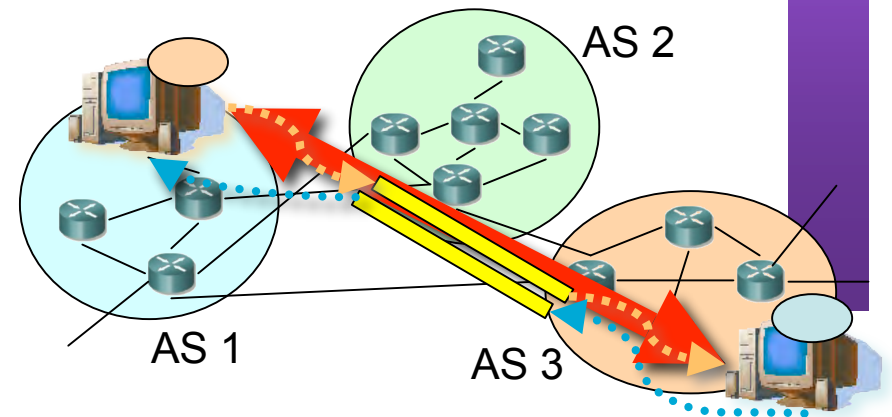
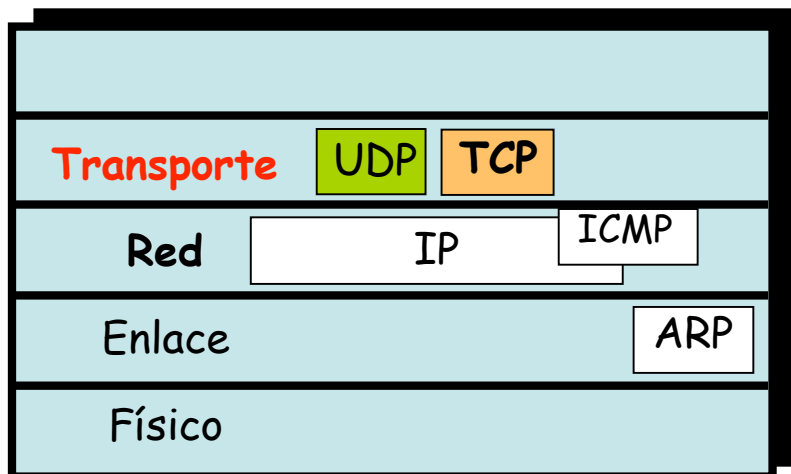
Contenido

- **Introducción**
- **Demultiplexación en TCP**
- Formato del segmento TCP
- Gestión de conexiones



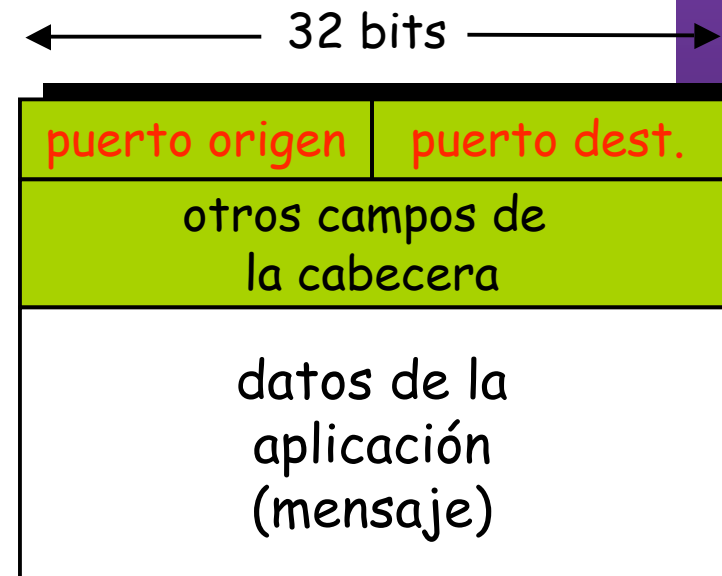
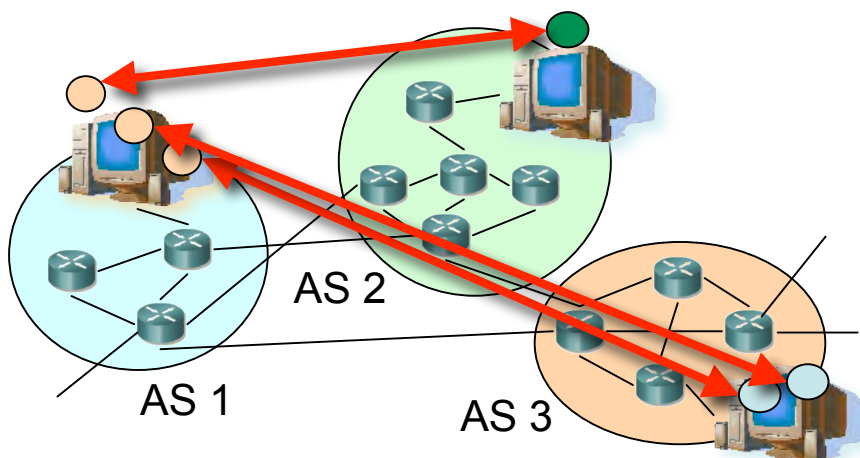
TCP

- *Transmission Control Protocol*
- Nivel de transporte
- RFCs 793, 1122, 1323, 2018, 2581
- Orientado a conexión
- Flujo de datos:
 - *Stream* de bytes
 - Fiable
 - Ordenado
 - Full duplex
- Control de flujo
 - Evitar congestionar al receptor
- Control de congestión
 - Evitar congestionar la red



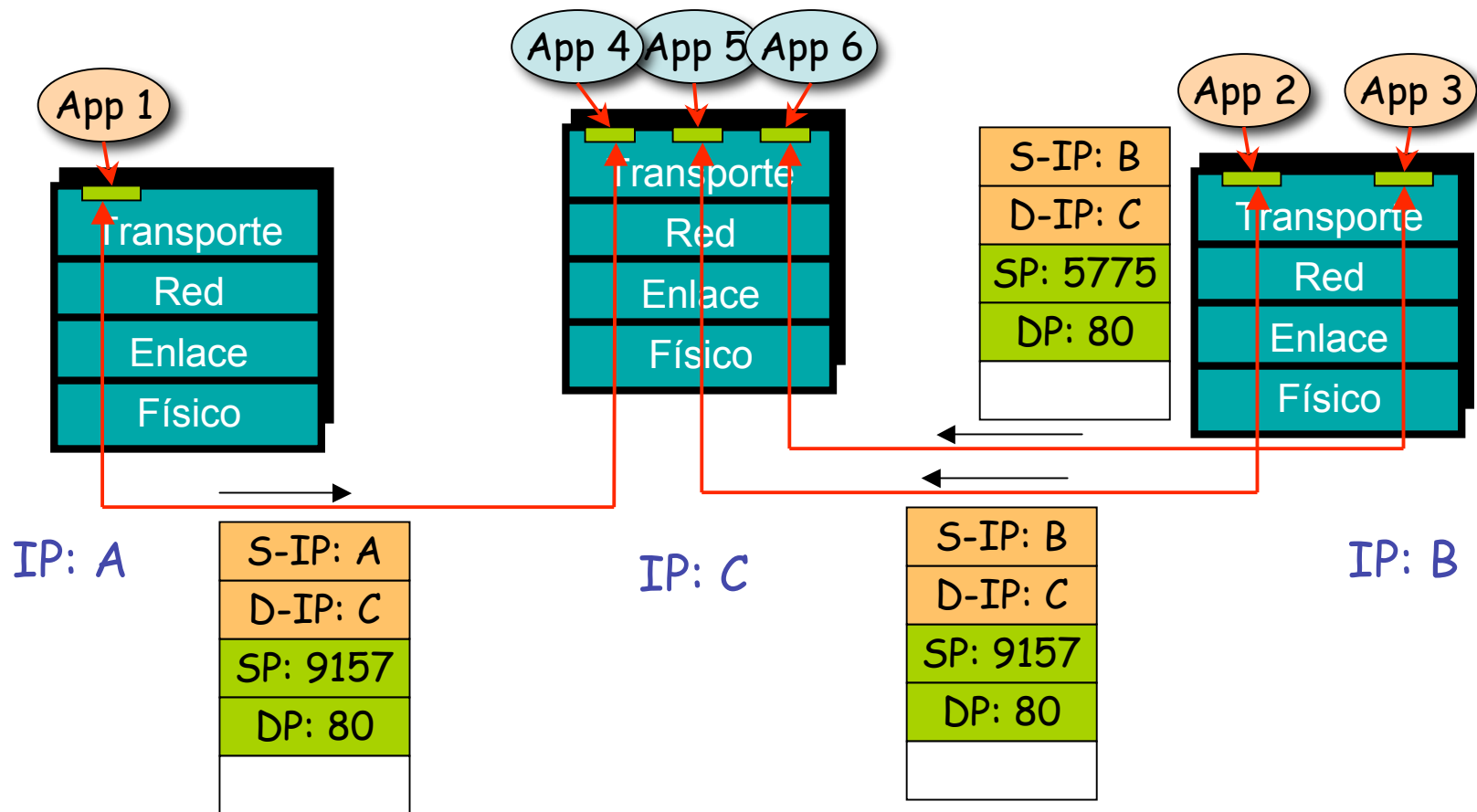
Demultiplexación con conexión

- Conexión identificada por 2 *sockets*
- Cada *socket* identificado por: Dirección IP y Puerto TCP
- Es decir, la conexión viene identificada por:
 - Dirección IP (1), Puerto TCP (1)
 - Dirección IP (2), Puerto TCP (2)
- El receptor emplea la cuaterna para demultiplexar
- Cada host soporta múltiples conexiones TCP simultáneas
- Con que uno de los 4 valores sea diferente la conexión ya es diferente
- *Well-known ports*, registrados, efímeros, igual que para UDP



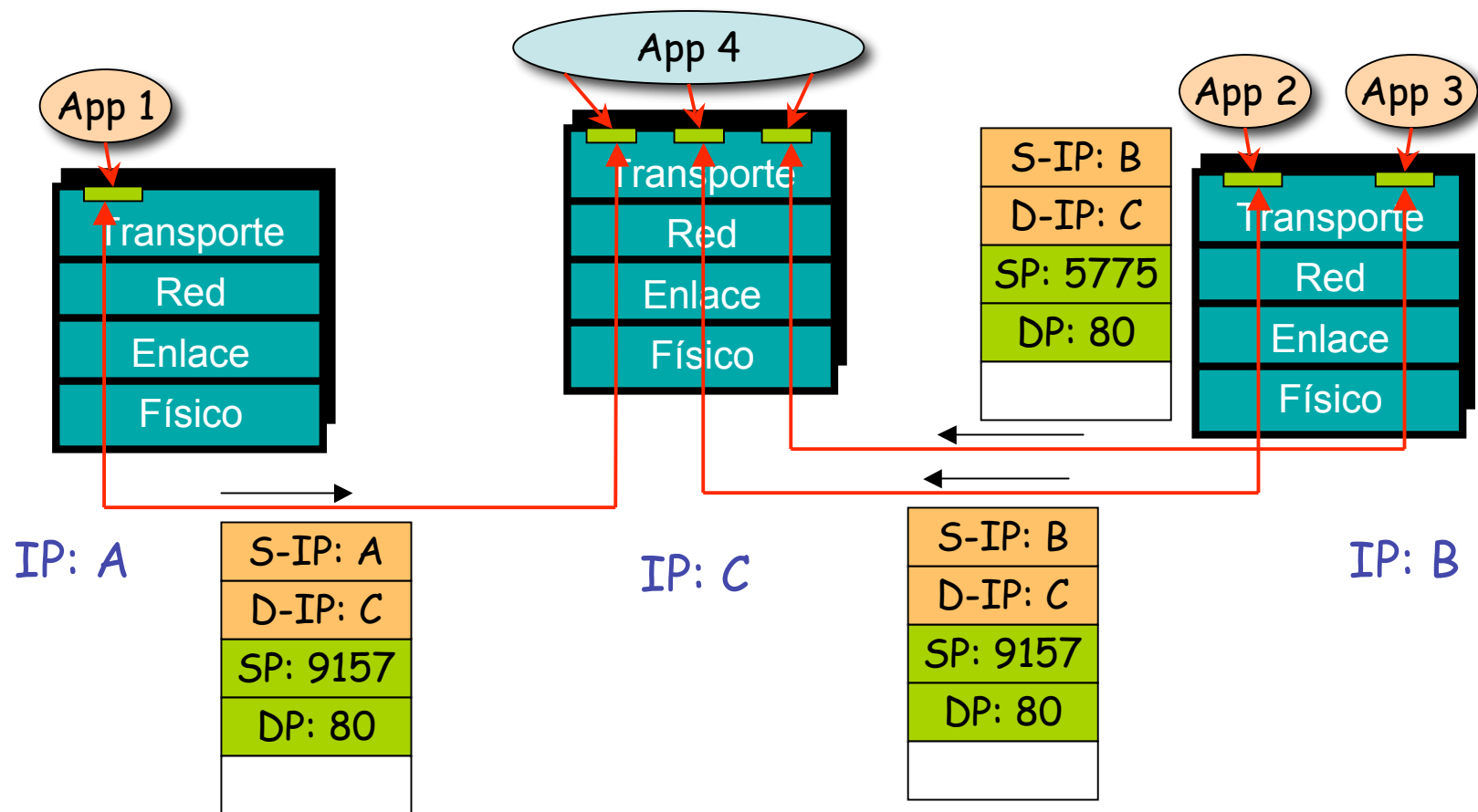


Demultiplexación: Ejemplo





Demultiplexación: otro ejemplo



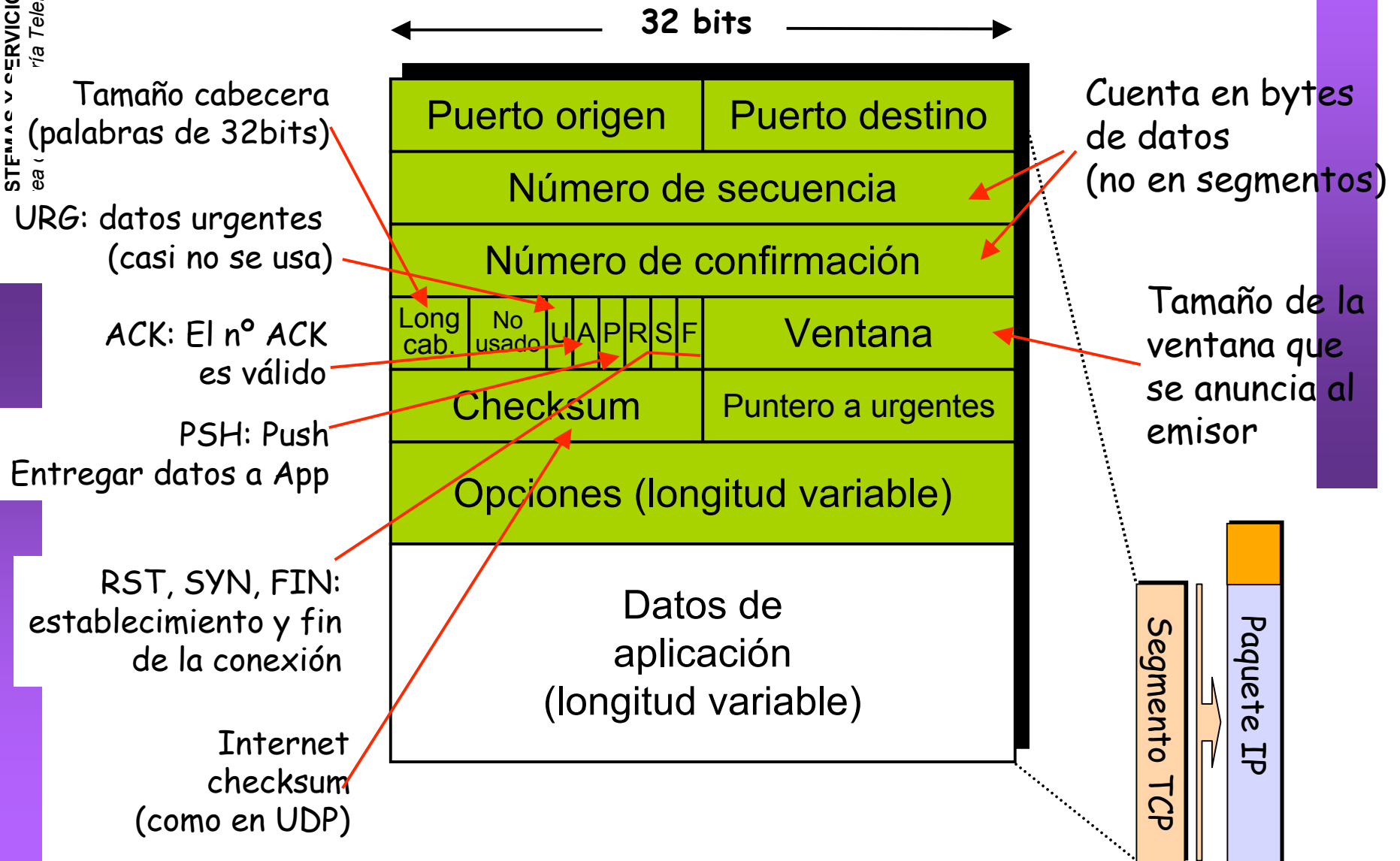


Contenido

- Introducción
- Demultiplexación en TCP
- **Formato del segmento TCP**
- Gestión de conexiones



Segmento TCP





Contenido

- Introducción
- Demultiplexación en TCP
- Formato del segmento TCP
- **Gestión de conexiones**



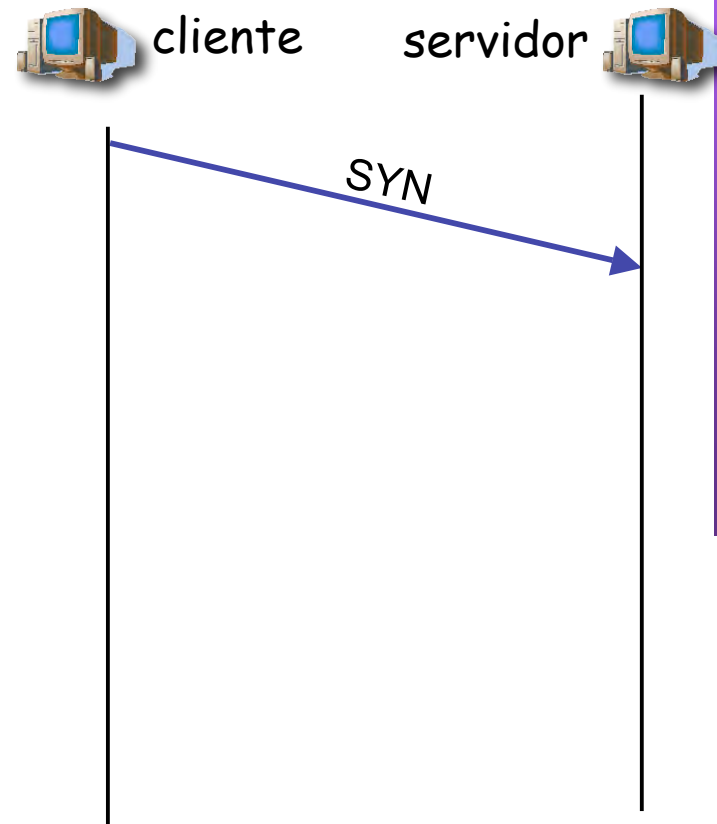
Gestión de conexiones

Estableciendo una conexión:

- *Three way handshake*

Paso 1:

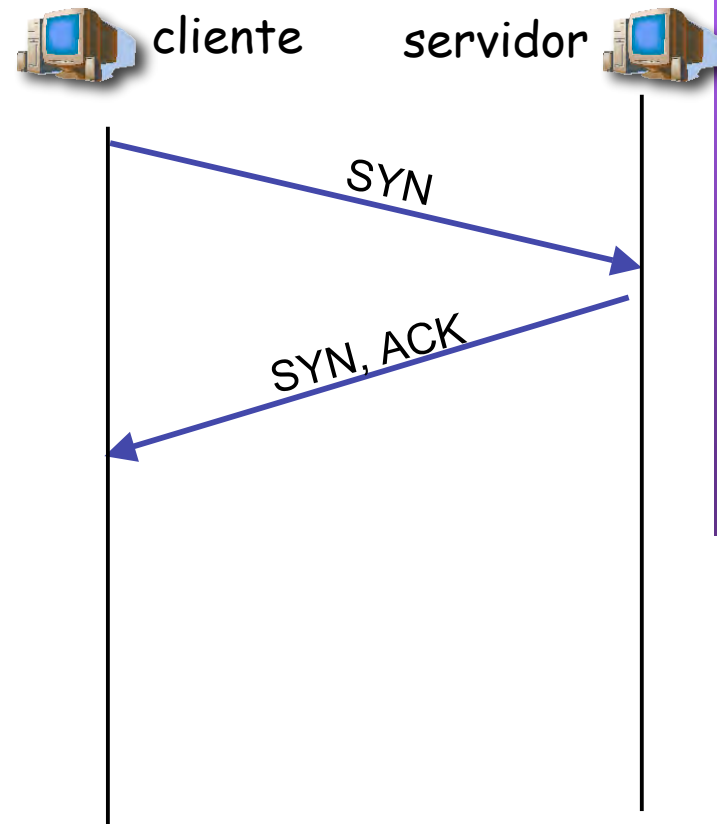
- El extremo **cliente** envía un segmento solicitando una conexión al servidor
- El segmento **no tiene datos**, solo cabecera
- **SYN**



Gestión de conexiones

Paso 2:

- El extremo **servidor** envía un segmento al cliente confirmando (acknowledgement) la recepción del SYN
- En el mismo segmento el servidor indica su deseo de establecer la conexión (SYN)
- El segmento **no tiene datos**, solo cabecera



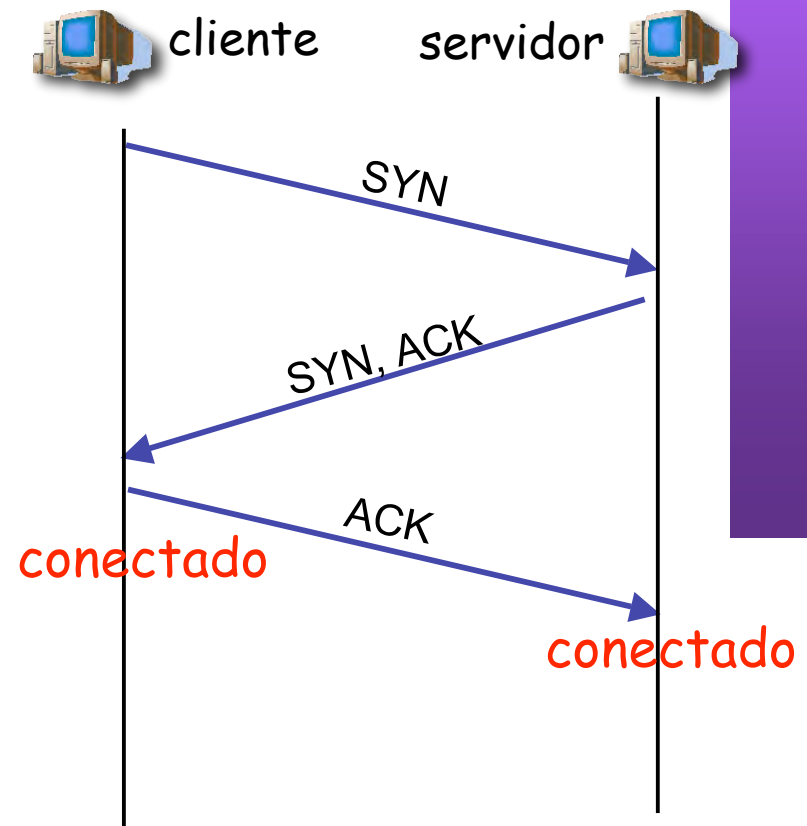


Gestión de conexiones

Paso 3:

- El extremo **cliente** envía una confirmación al SYN del servidor
- El segmento **no tiene datos**, solo cabecera
- Conexión establecida

Transferencia de datos...



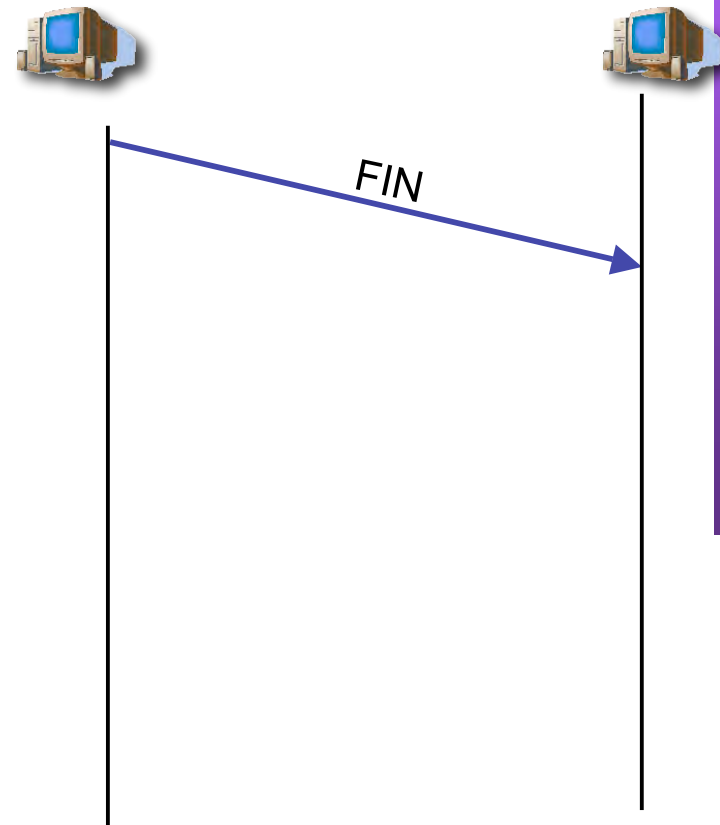


Gestión de conexiones

Cerrando una conexión

Paso 1:

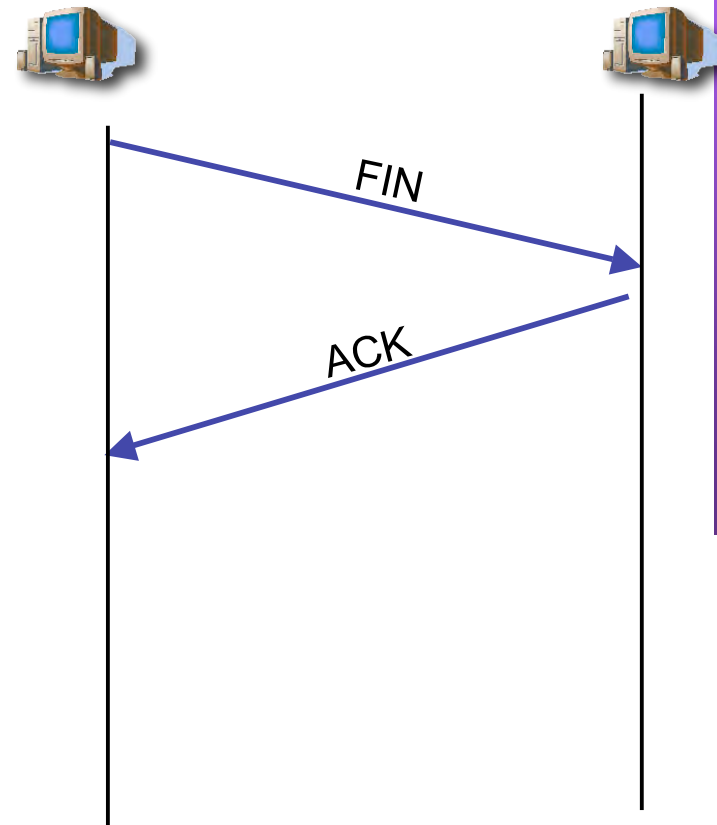
- Un extremo envía un segmento solicitando el cierre de la conexión
- El segmento **no tiene datos**, solo cabecera
- **FIN**



Gestión de conexiones

Paso 2:

- El otro extremo confirma (ACK) la recepción del FIN
- El extremo que ha enviado el FIN ya no puede enviar más datos nuevos
- **Cierre solo de un sentido** de la comunicación

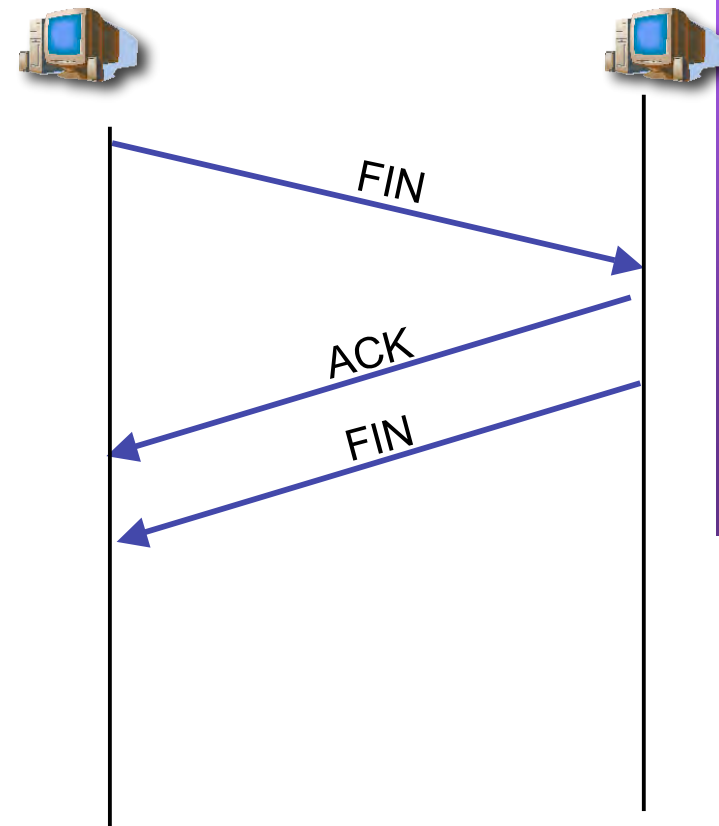




Gestión de conexiones

Paso 3:

- El otro extremo envía un segmento solicitando el cierre de la conexión
- El segmento no tiene datos, solo cabecera





Gestión de conexiones

Paso 4:

- Confirmación de ese segundo FIN
- Por si ese último ACK se pierde, el que lo envió espera un tiempo (podría tener que volverlo a enviar)
- Conexión cerrada

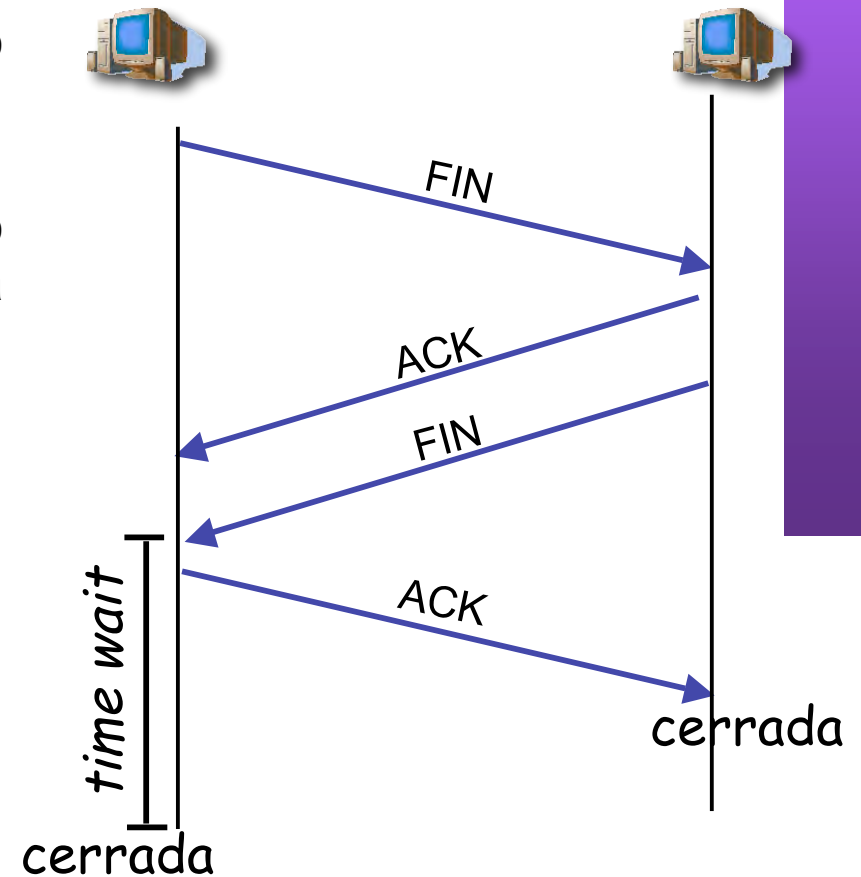




Diagrama de estados

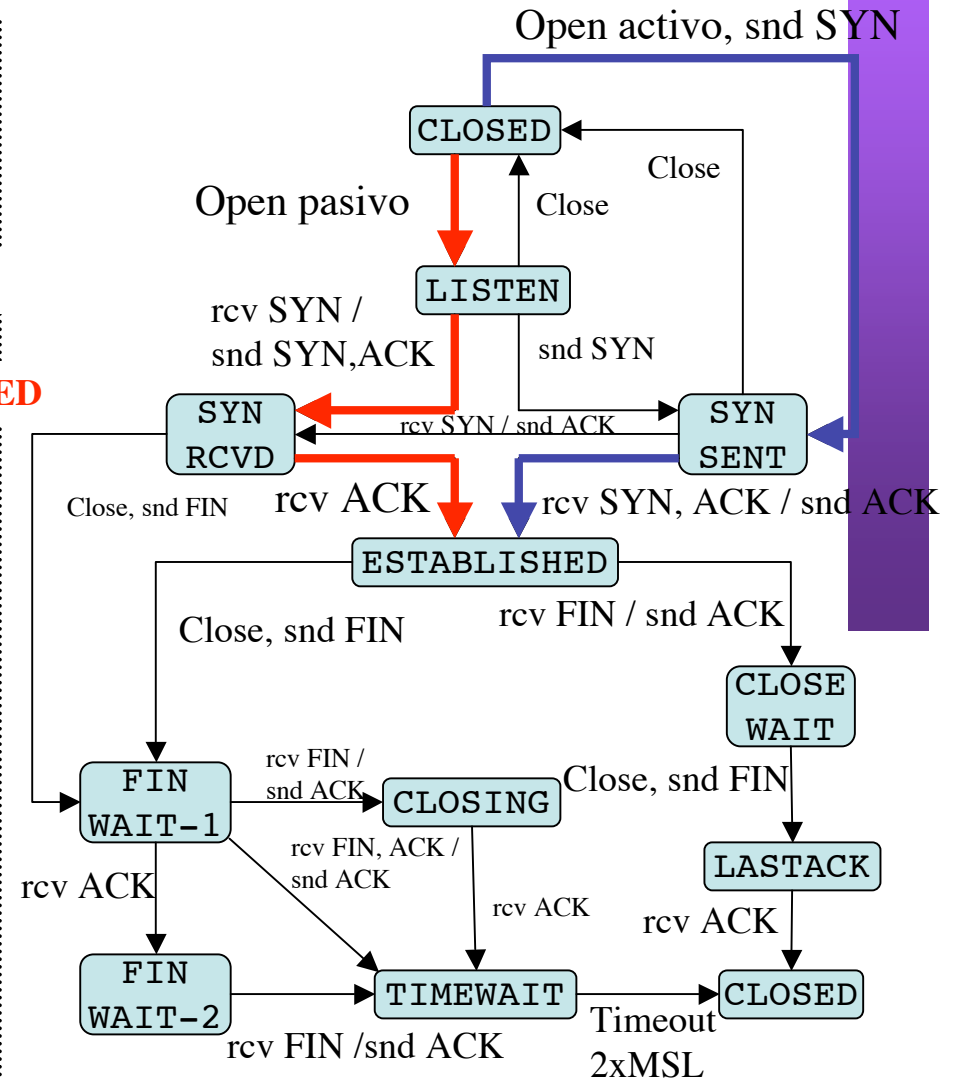
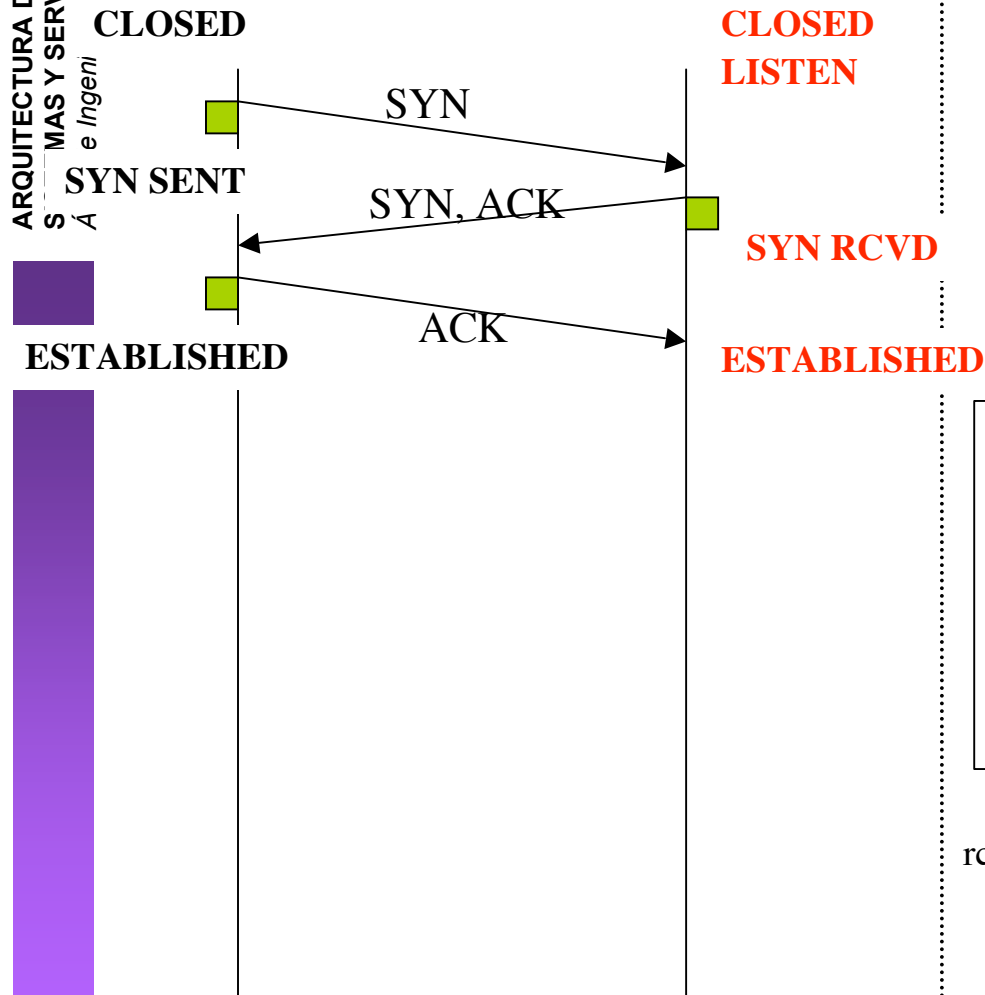




Diagrama de estados

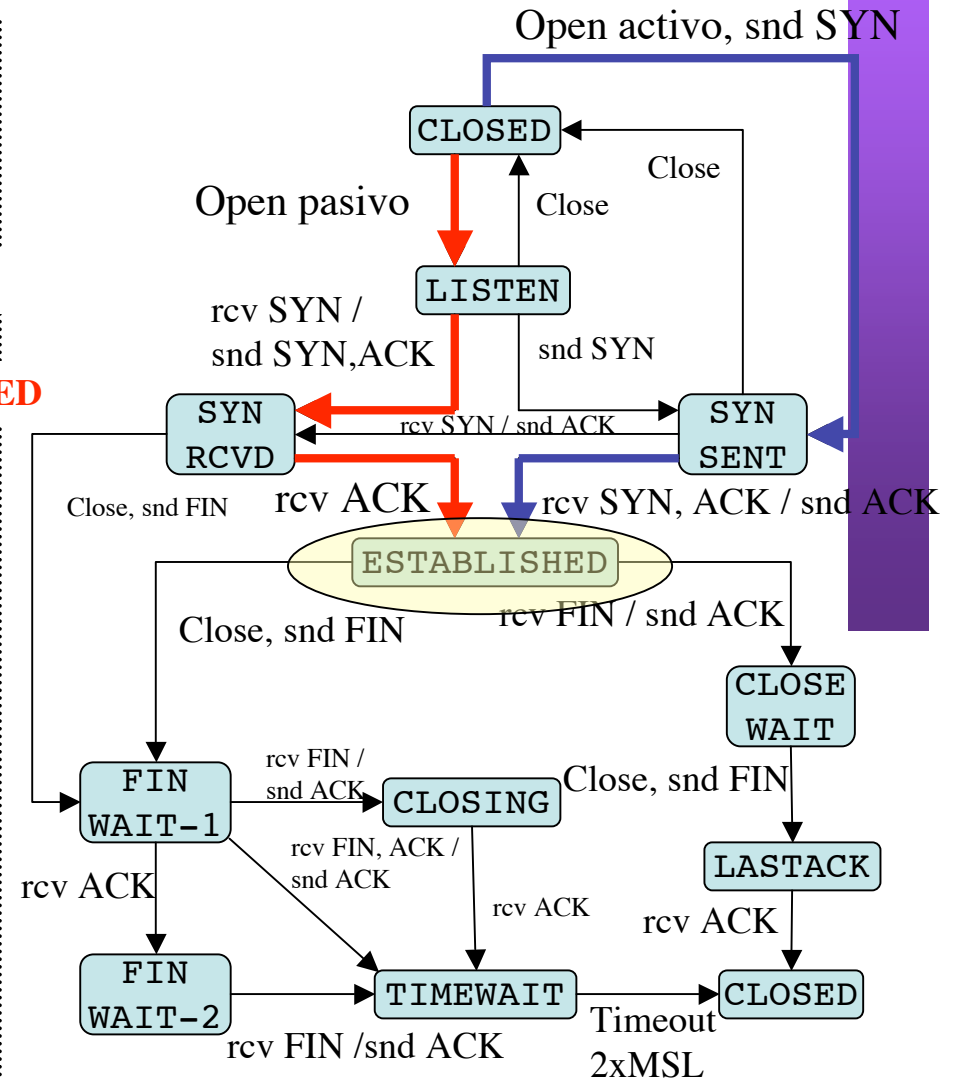
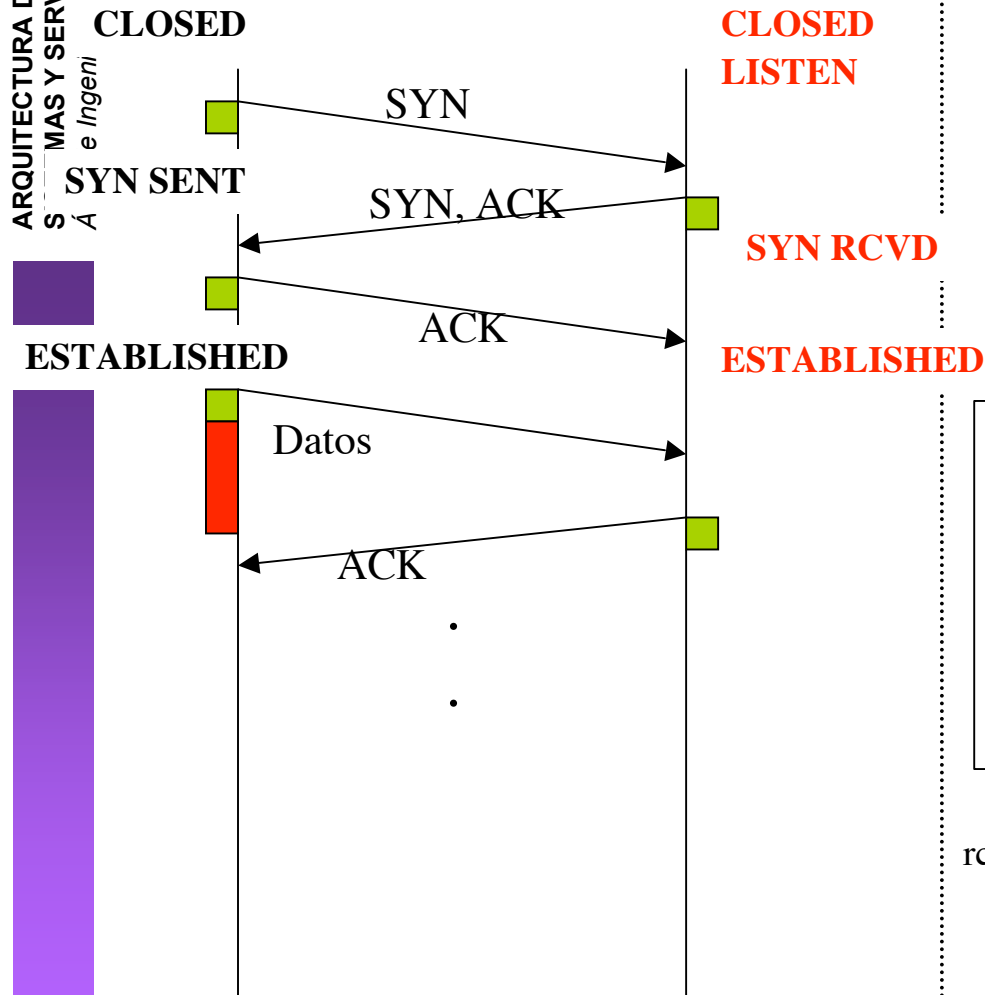
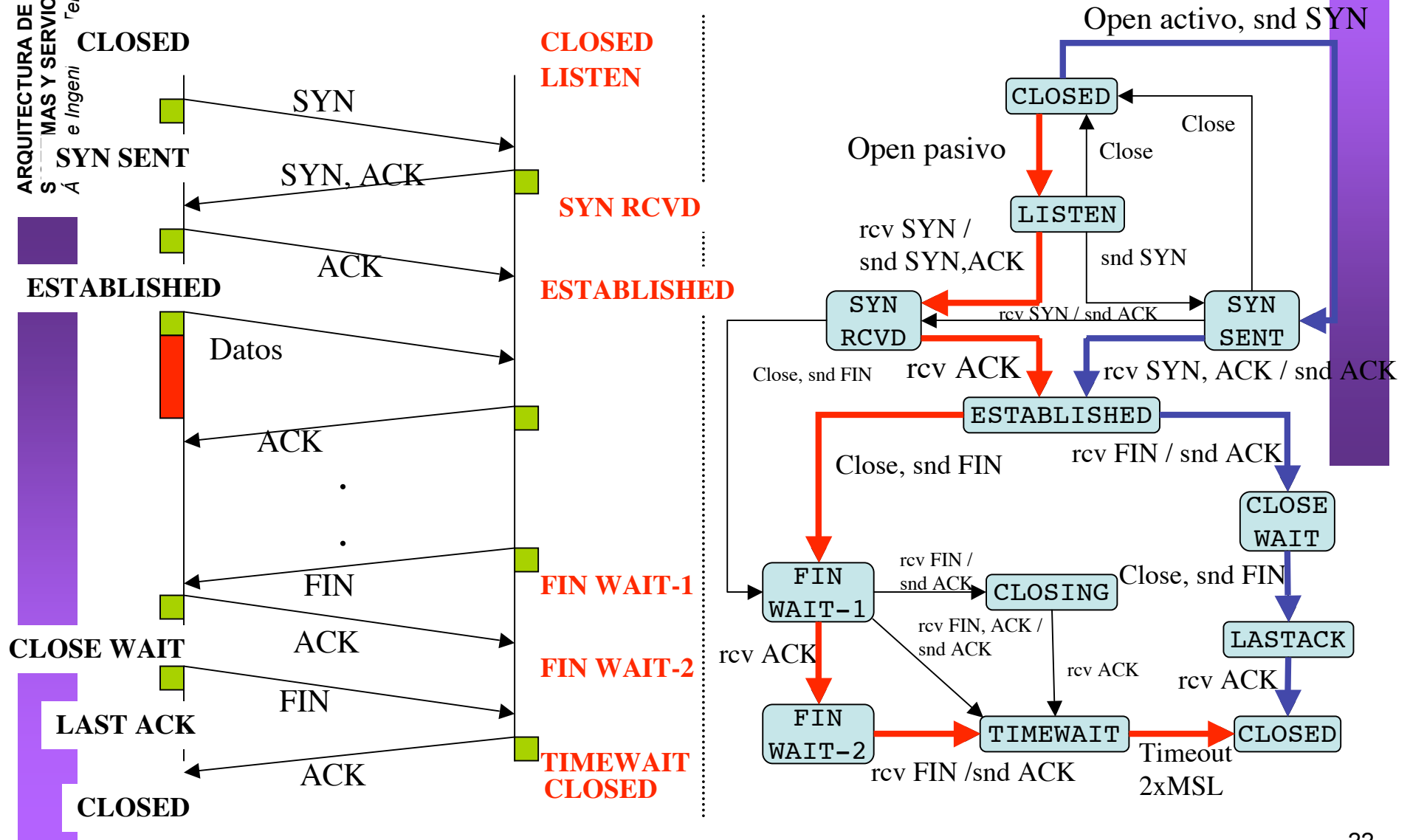


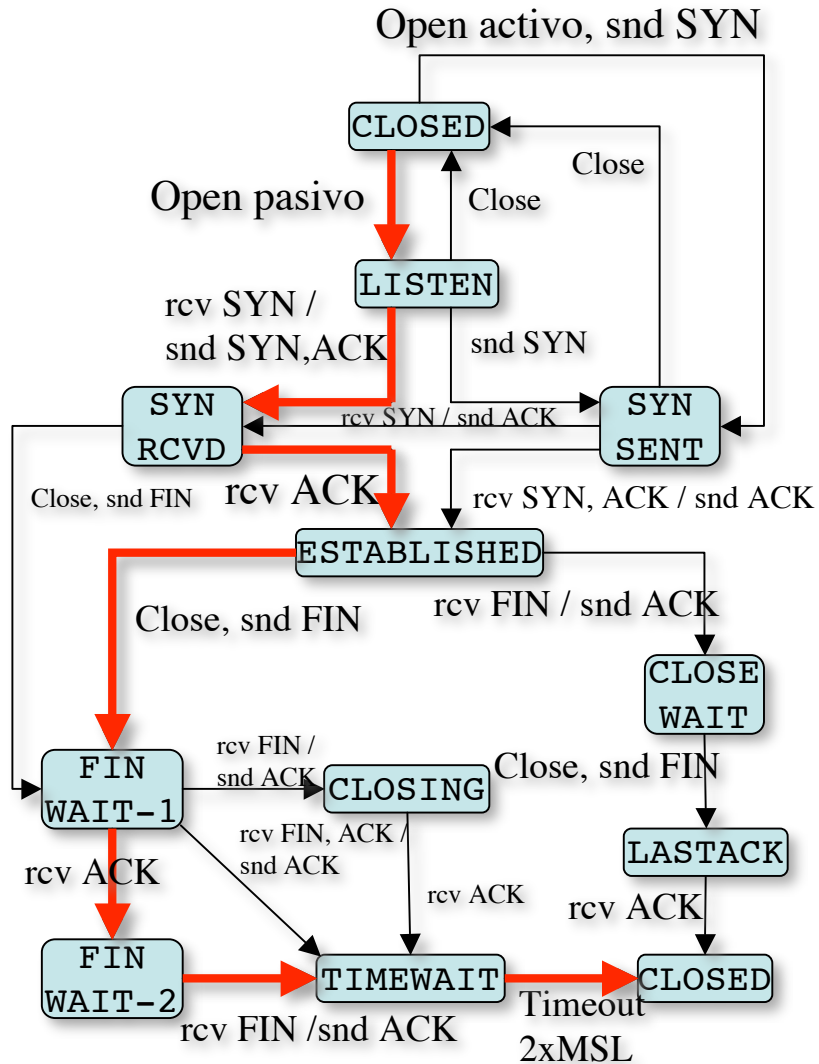


Diagrama de estados

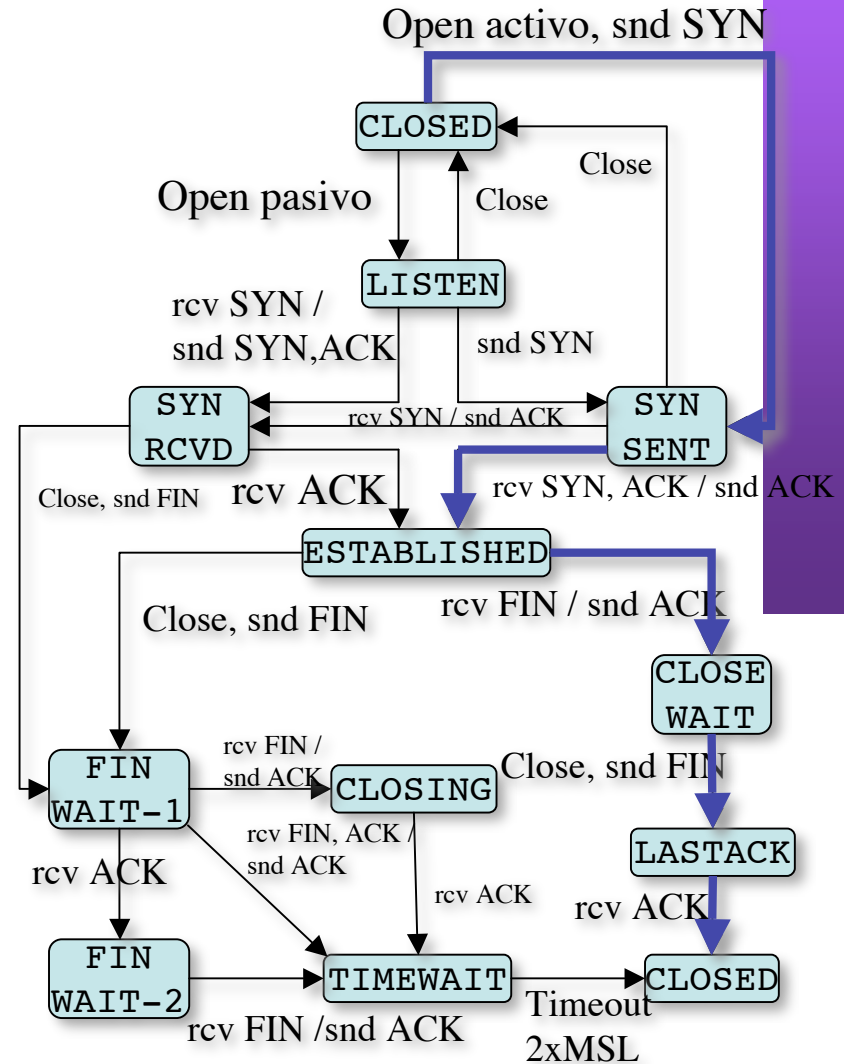




Servidor



Cliente





Ejemplo

```
$ tcpdump -ttnls tcp and host 10.1.11.1
Kernel filter, protocol ALL, datagram packet socket
tcpdump: listening on all devices
54.171 1.1.1.12.1798 > 10.1.11.1.telnet: S 3462181145:3462181145(0)
54.175 10.1.11.1.telnet > 1.1.1.12.1798: S 1997882026:1997882026(0) ack 3462181146
54.175 1.1.1.12.1798 > 10.1.11.1.telnet: . 3462181146:3462181146(0) ack 1997882027

54.177 1.1.1.12.1798 > 10.1.11.1.telnet: P 3462181146:3462181173(27) ack 1997882027
54.178 10.1.11.1.telnet > 1.1.1.12.1798: . 1997882027:1997882027(0) ack 3462181173
...

66.816 10.1.11.1.telnet > 1.1.1.12.1798: FP 1997882551:1997882559(8) ack 3462181333
66.816 1.1.1.12.1798 > 10.1.11.1.telnet: . 3462181333:3462181333(0) ack 1997882560
66.817 1.1.1.12.1798 > 10.1.11.1.telnet: F 3462181333:3462181333(0) ack 1997882560
66.818 10.1.11.1.telnet > 1.1.1.12.1798: . 1997882560:1997882560(0) ack 3462181334
```



Resumen

- Fiable, mantiene el orden, flujo en stream...
- TCP emplea el concepto de conexión
- *Three-way handshake*
- (IP_1, puerto_1, IP_2, puerto_2)
- Mantiene estado



Próxima clase

Servicios en Internet: WWW

- Lecturas recomendadas:
 - [Kurose05] 2.2