Problema de agotamiento de direcciones

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

Laboratorio de Programación de Redes
3º Ingeniería Técnica en Informática de Gestión

Objetivo

- Ver diferentes soluciones al problema de la escasez de direcciones IP

Contenido

- Introducción
- El problema
- Algunas soluciones
  - DHCP
  - NAT
  - IPv6

Contenido

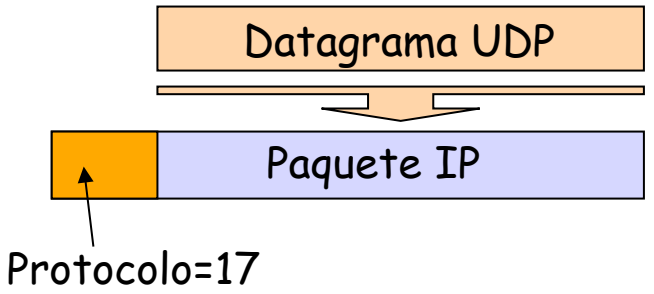
- **Introducción**
- El problema
- Algunas soluciones
  - DHCP
  - NAT
  - IPv6

UDP: User Datagram Protocol

- RFC 768
- Protocolo de transporte simple, sin gran inteligencia
- Servicio “best effort”
- Datagramas
- Los datagramas UDP se pueden:
  - Perder
  - Llegar desordenados a la aplicación
- ¿Transferencia fiable sobre UDP?
  - Añadir fiabilidad en el nivel de aplicación
  - ¡Recuperación ante errores específica de cada aplicación!
- Sin conexión:
  - No hay handshaking entre emisor y receptor
  - Cada datagrama UDP es procesado de forma independiente a los demás
- Empleado frecuentemente para aplicaciones de streaming multimedia
  - Soportan pérdidas
  - Sensibles a la tasa de envío
- Otros usos de UDP:
  - DNS
  - SNMP

UDP: User Datagram Protocol

- ¿Por qué existe UDP?
  - Es simple: no hay que mantener estado
  - Un establecimiento de conexión añadiría retardo no deseado
  - Cabecera pequeña
  - No hay control de congestión: puede enviar tan rápido como desee
- Encapsulado en paquetes IP, protocolo 17
- Cuando un host recibe un datagrama UDP :
  - Comprueba el puerto destino en el mismo
  - Dirige el segmento a la aplicación esperando datos a ese puerto
- Diferentes IP origen o puertos origen van al mismo punto de acceso al servicio (SAP)



Cabecera UDP

Puerto origen

- Normalmente lo escoge el sistema operativo
- Suele ser un puerto efímero

Puerto destino

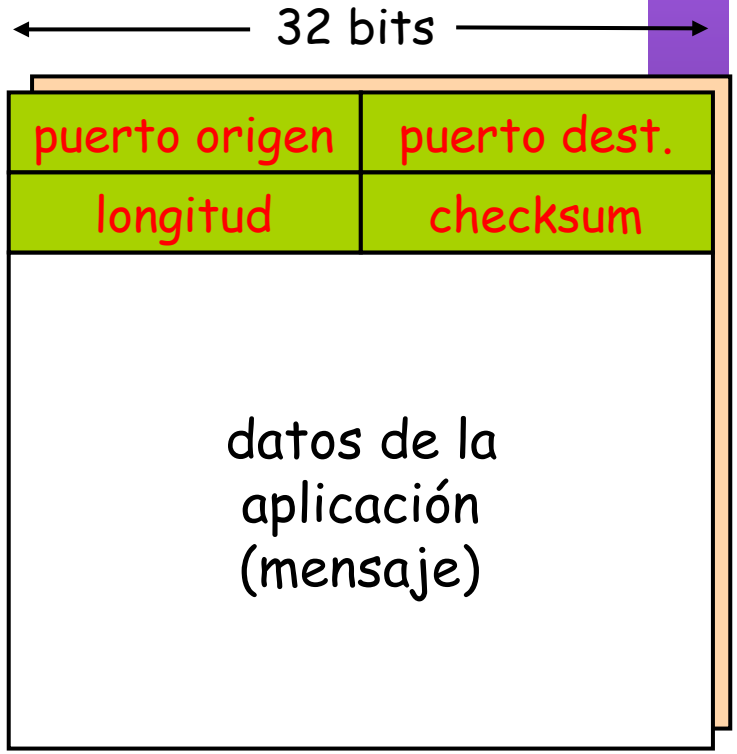
- Puerto del servidor
- *Well known* o se debe conocer por algún medio

Respuesta servidor→cliente

- Sentido contrario
- Puerto origen es el del servidor (*well known*)
- Puerto destino el efímero del cliente

Longitud

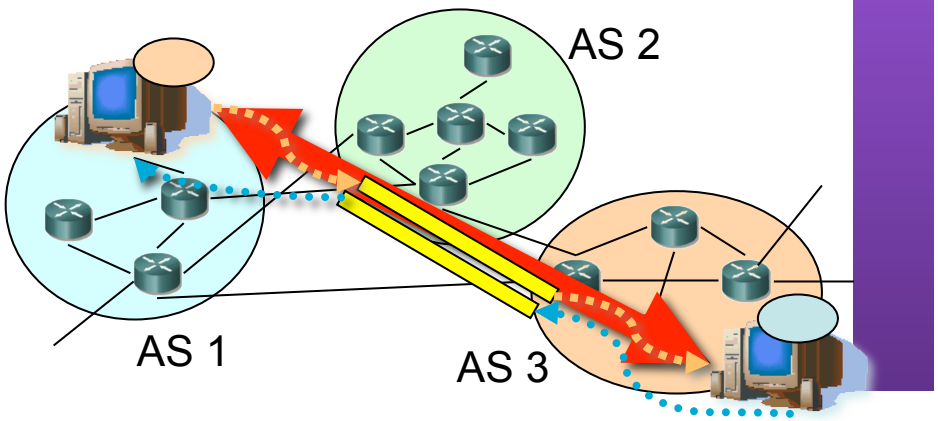
- Bytes del datagrama UDP

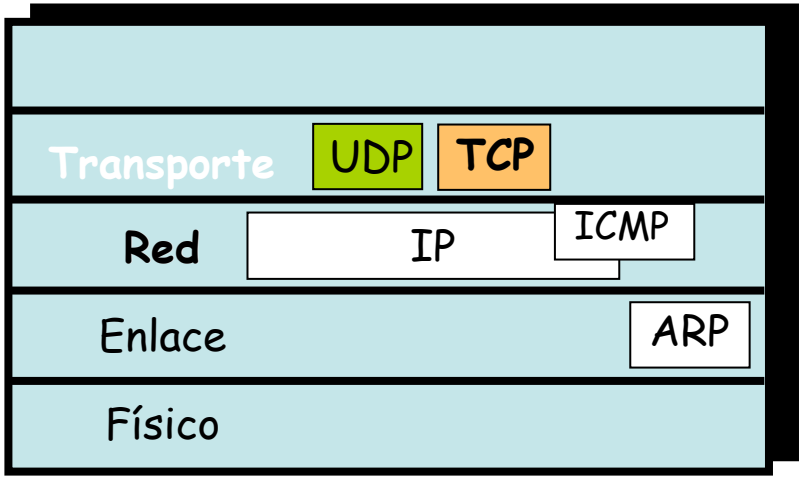


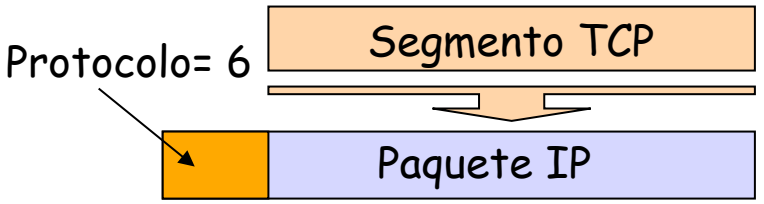
TCP

- *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

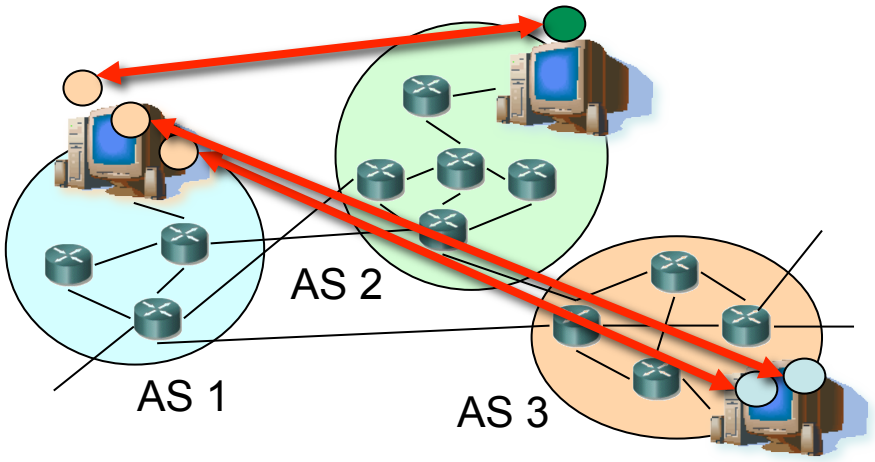


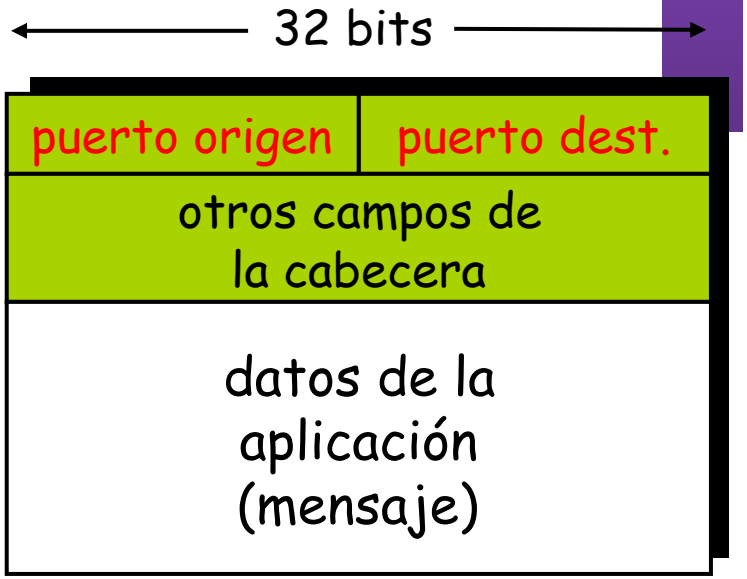




Demultiplexación con conexión

- 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





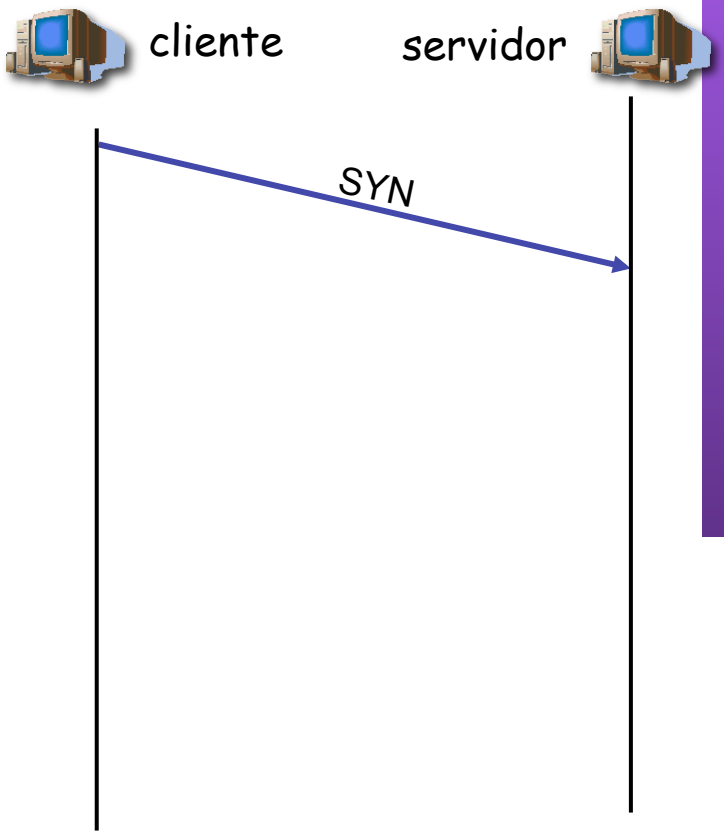
Gestión de conexiones

Estableciendo una conexión:

- *Three way handshake*

Paso 1:

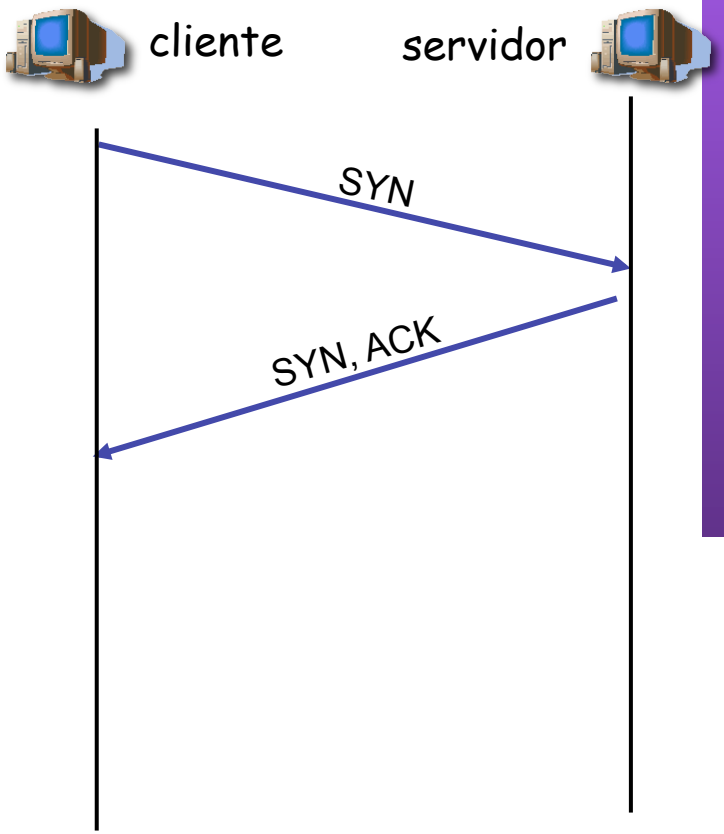
- El extremo **cliente** envía un segmento solicitando una conexión al servidor
- El segmento **no tiene datos**, solo cabecera
- **SYN**



Gestión de conexiones

Paso 2:

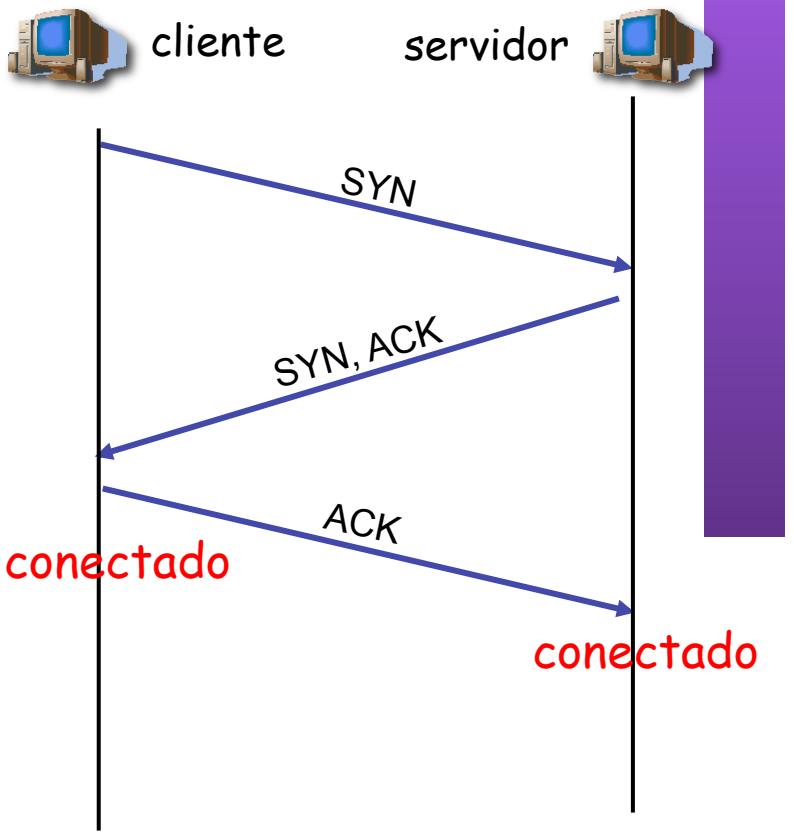
- El extremo **servidor** envía un segmento al cliente confirmando (acknowledgement) la recepción del SYN
- En el mismo segmento el servidor indica su deseo de establecer la conexión (SYN)
- El segmento **no tiene datos**, solo cabecera



Gestión de conexiones

Paso 3:

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



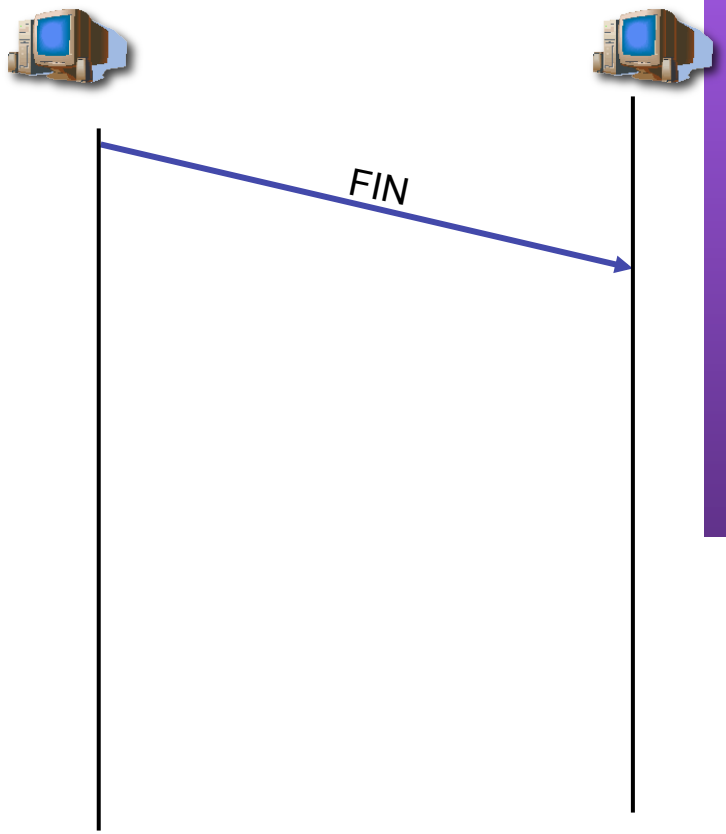
Transferencia de datos…

Gestión de conexiones

Cerrando una conexión

Paso 1:

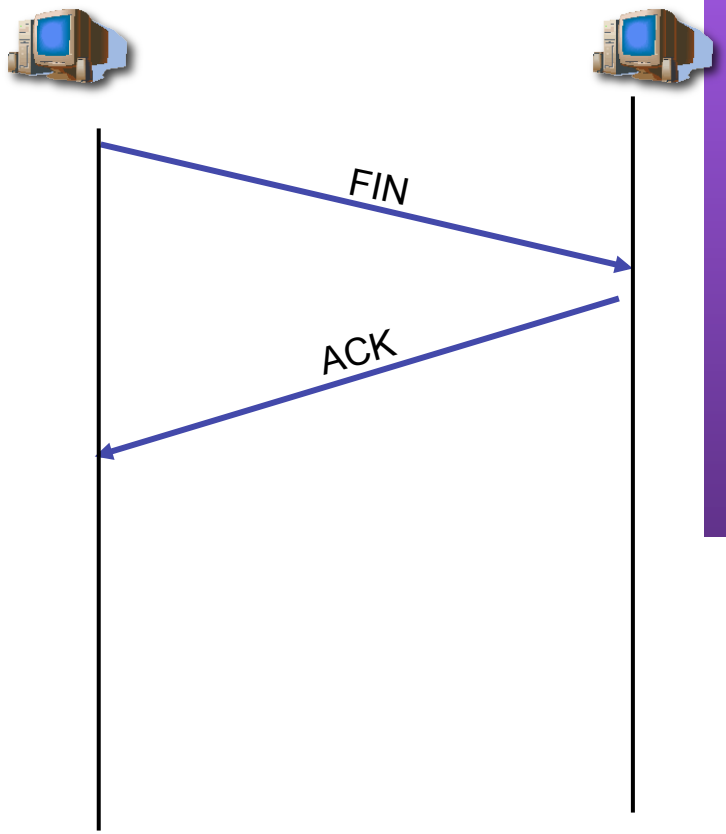
- **Un extremo** envía un segmento solicitando el cierre de la conexión
- El segmento **no tiene datos**, solo cabecera
- **FIN**



Gestión de conexiones

Paso 2:

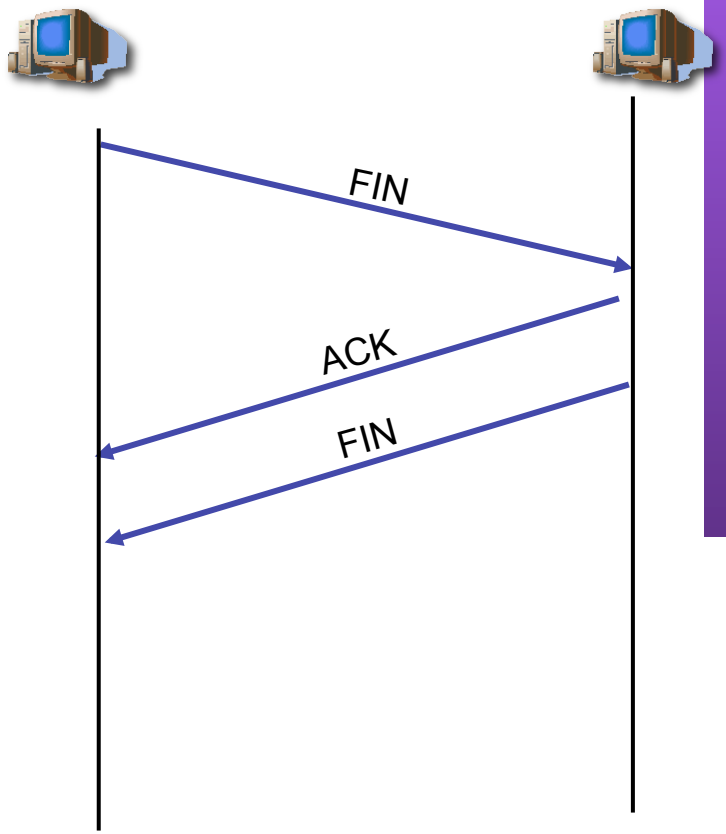
- El otro extremo confirma (ACK) la recepción del FIN
- El extremo que ha enviado el FIN ya no puede enviar más datos nuevos
- **Cierre solo de un sentido** de la comunicación



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

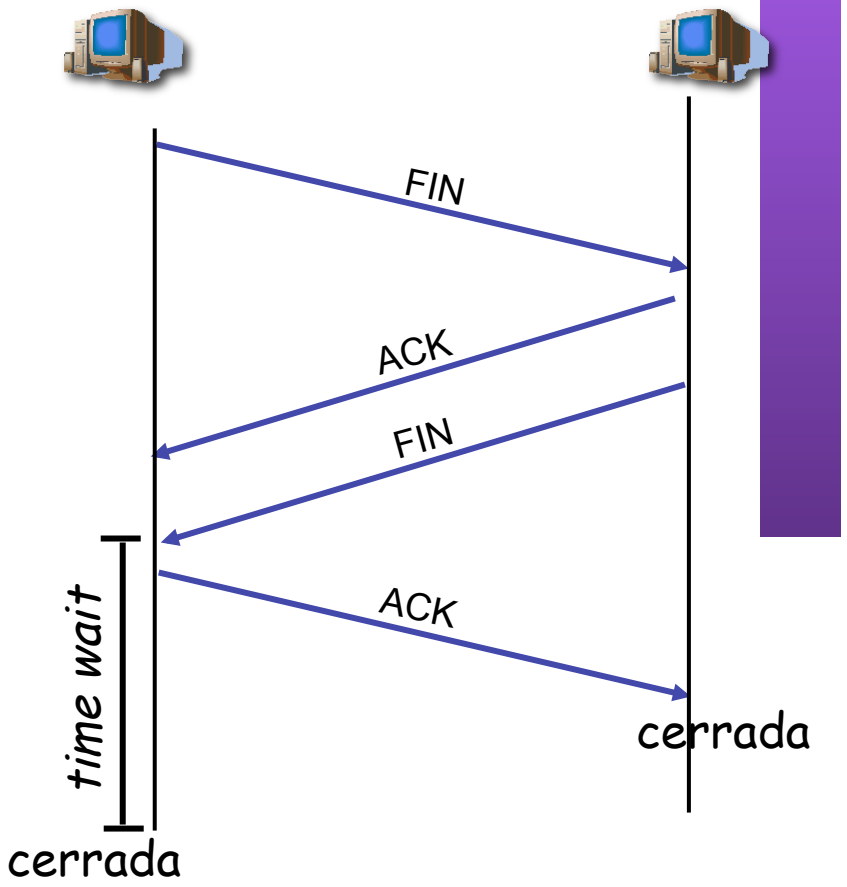
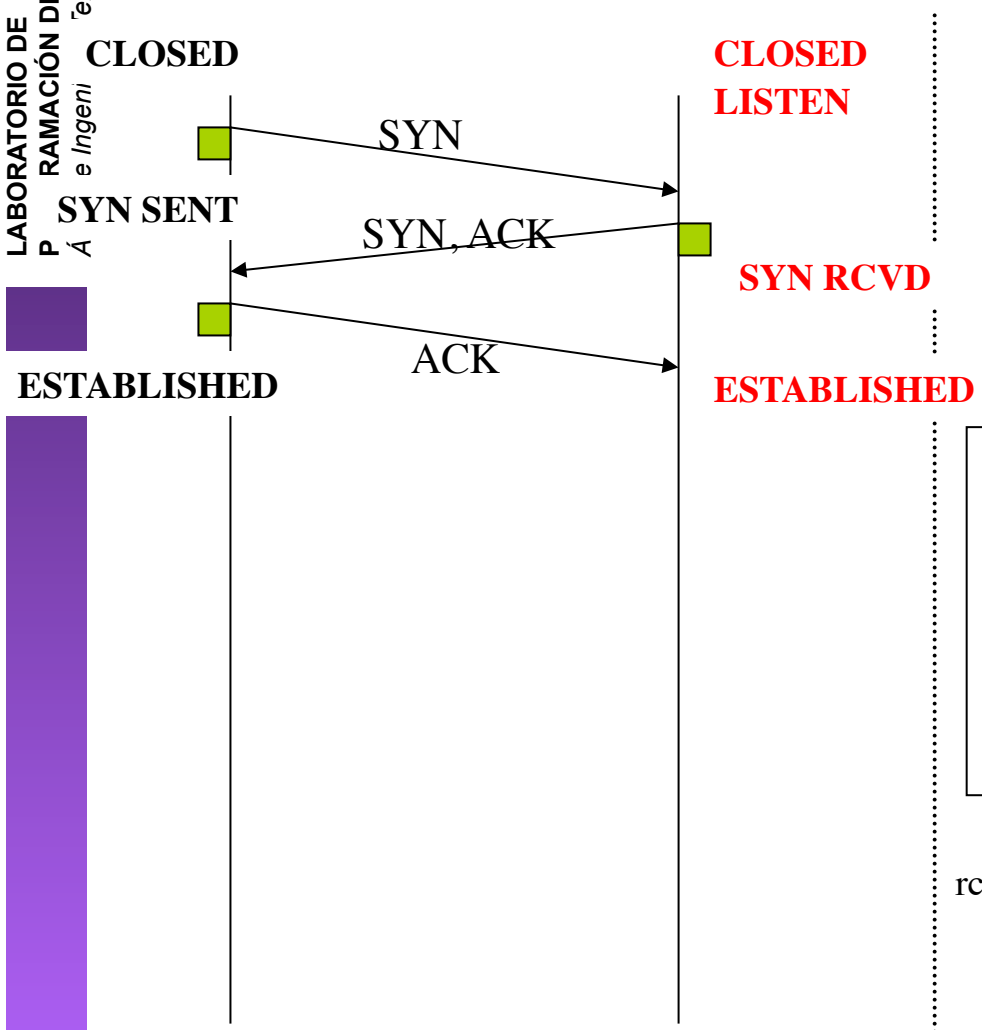


Diagrama de estados



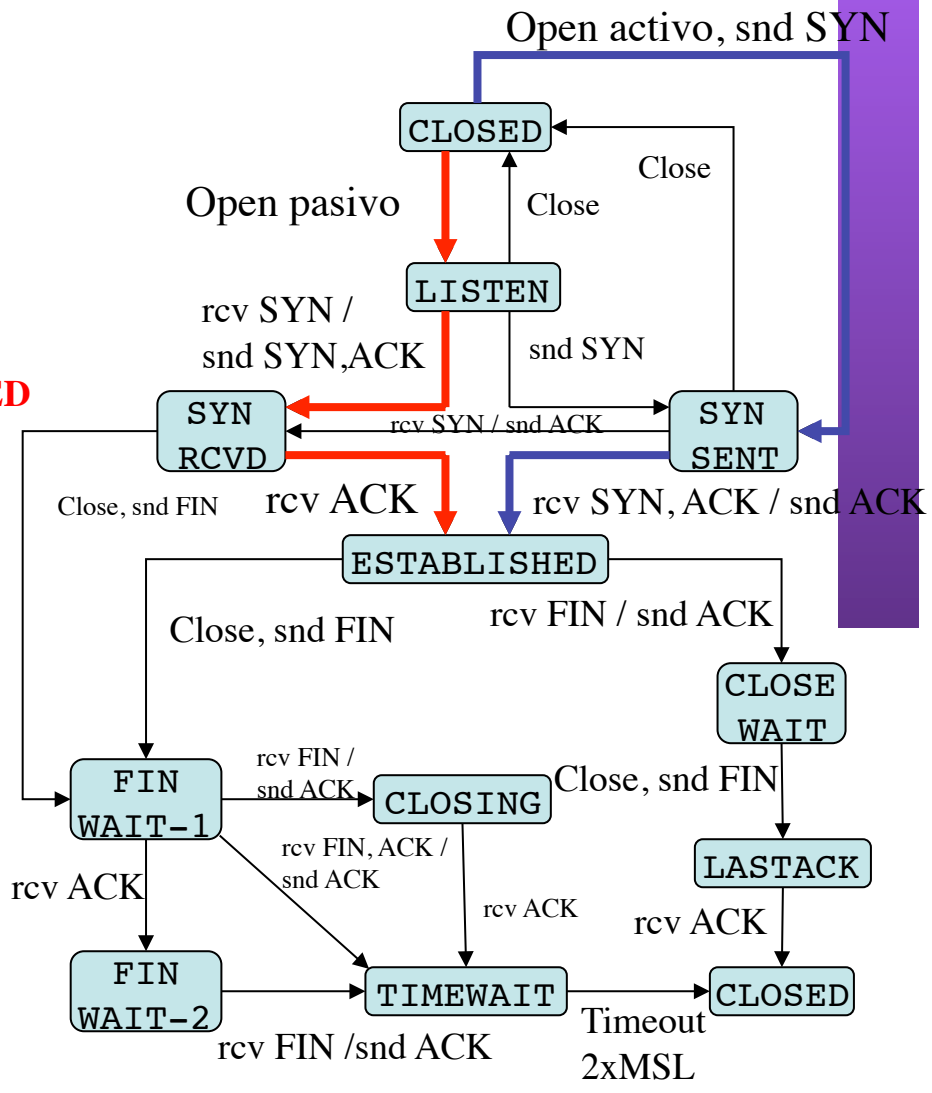
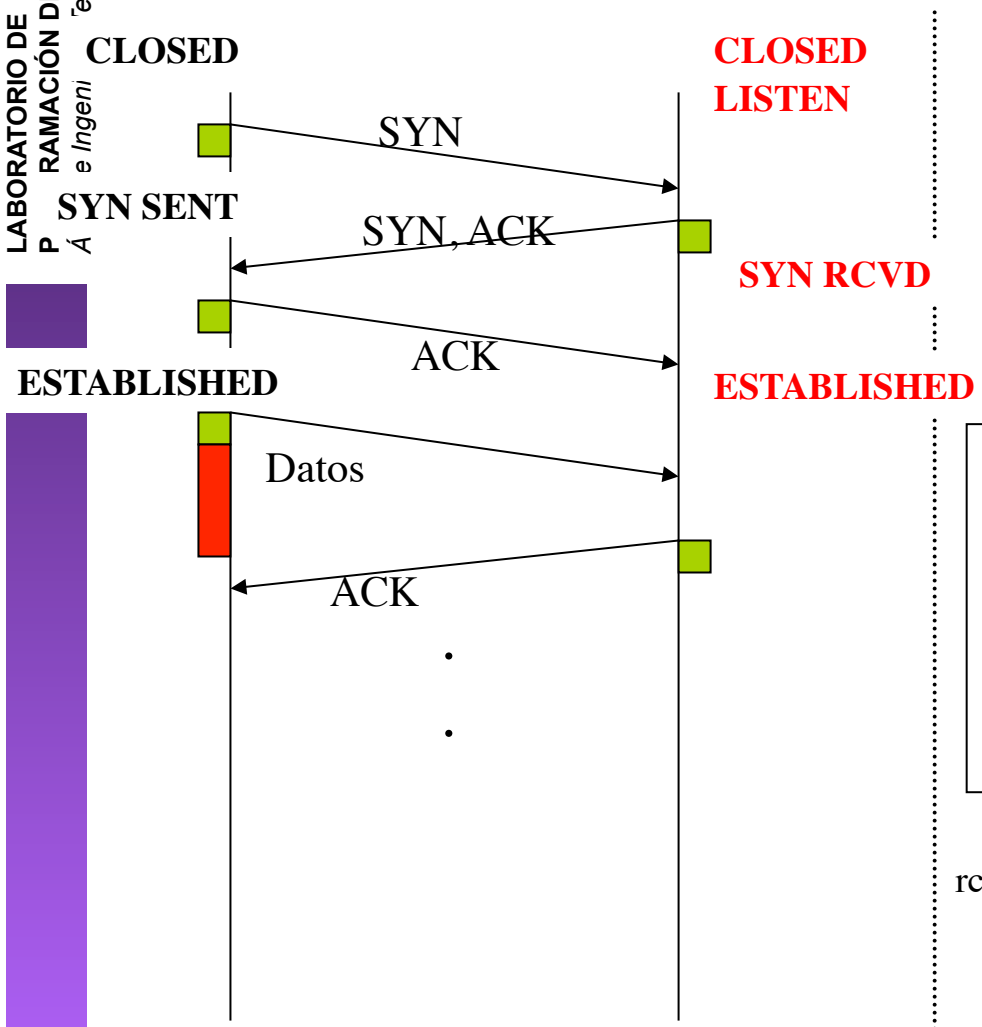


Diagrama de estados



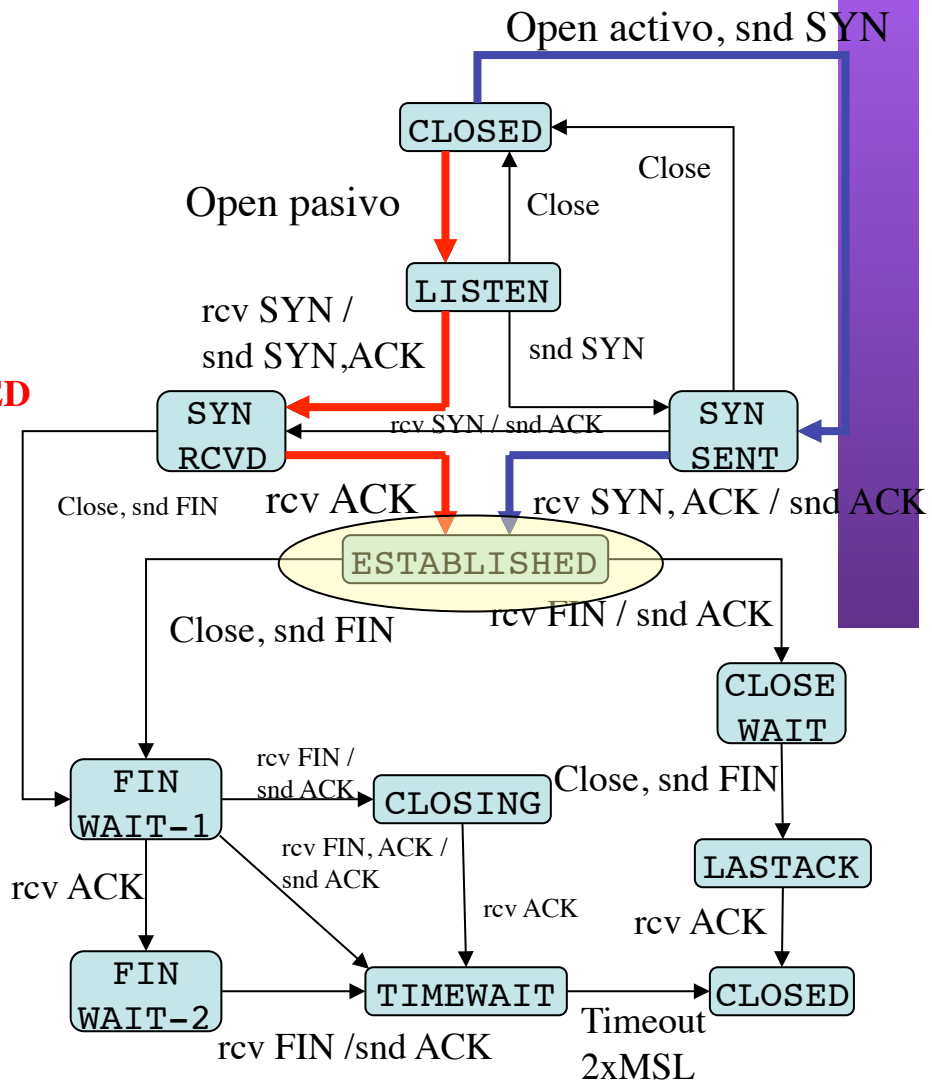
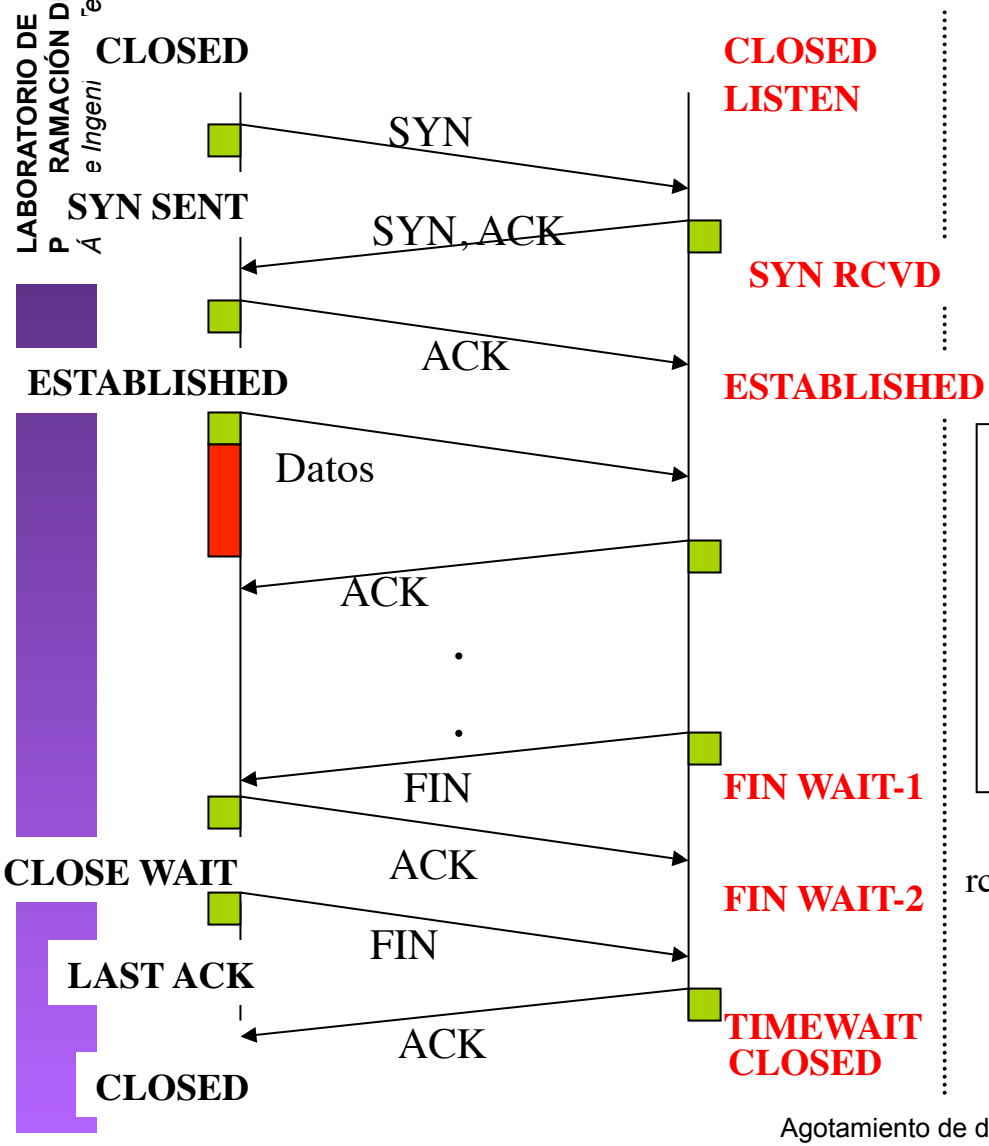
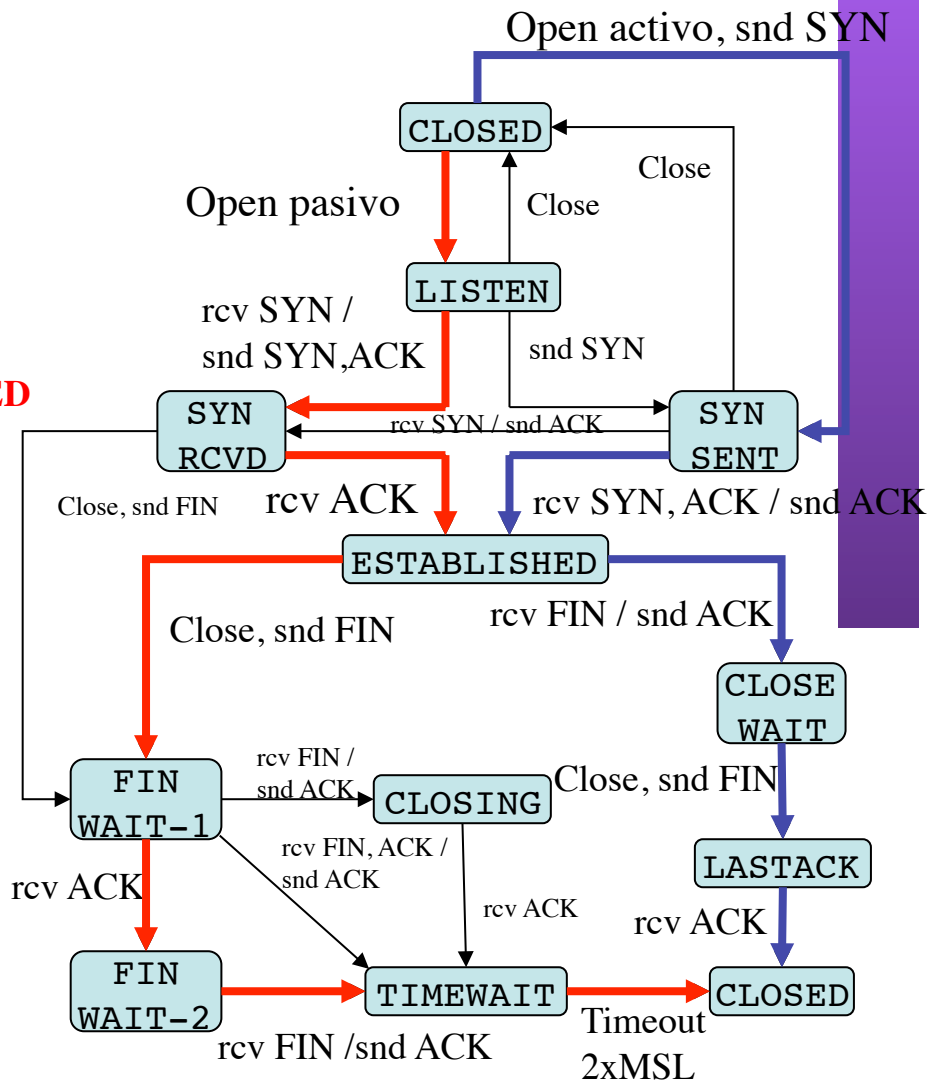
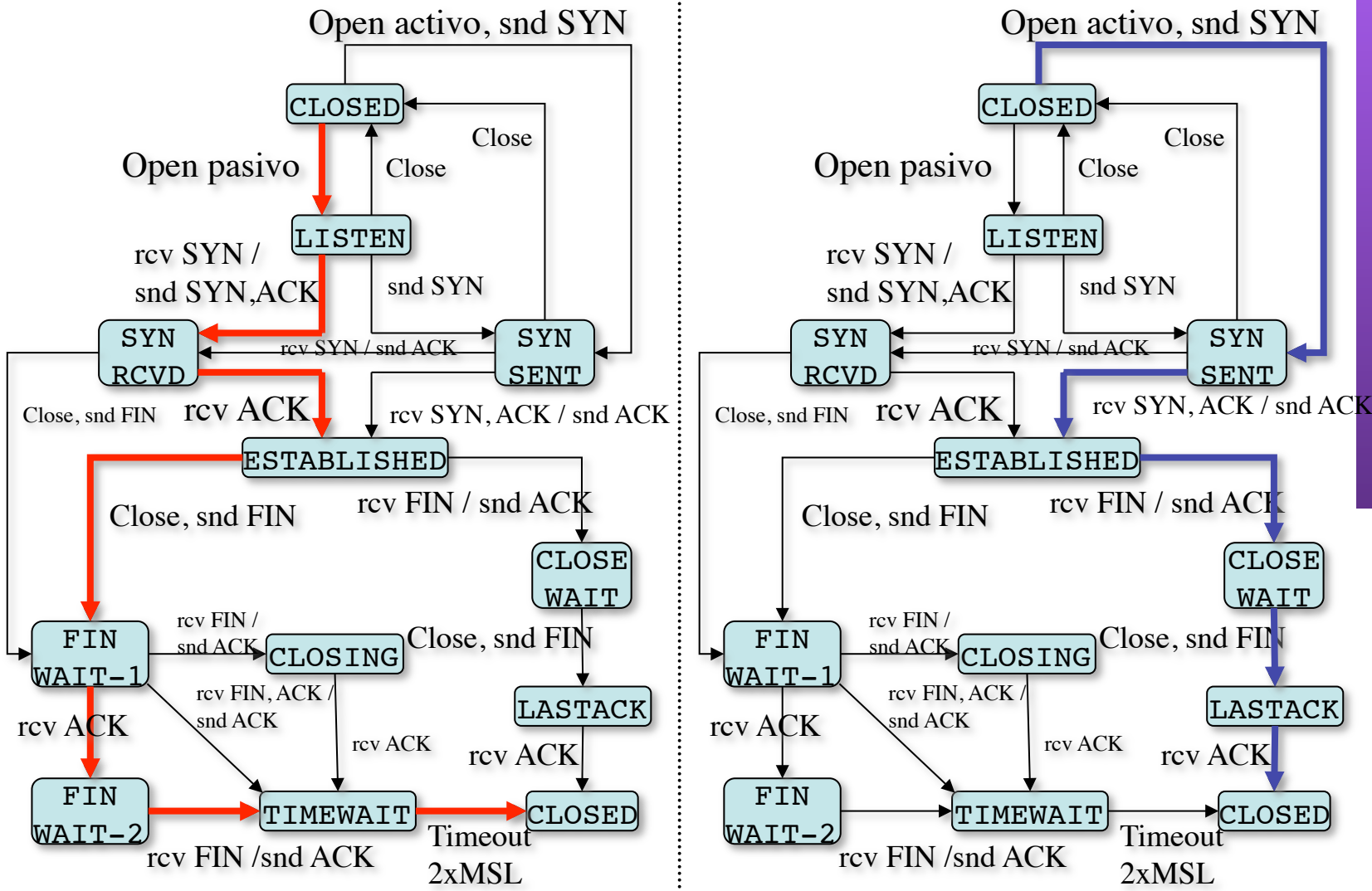


Diagrama de estados





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
```

Contenido

- Introducción
- El problema
- Algunas soluciones
  - DHCP
  - NAT
  - IPv6

Contenido

- Introducción
- **El problema**
- Algunas soluciones
  - DHCP
  - NAT
  - IPv6

Problemas de IPv4

- Escasez de direcciones
- Complejidad innecesaria en los routers

¿Dónde se desperdician direcciones?

- Redes con clases:
  - Clase A: Más de 16M de direcciones
  - Clase B: 64K direcciones
- PCs que se usen esporádicamente

A continuación…

- Introducción
- El problema
- **Algunas soluciones**
  - **DHCP**
  - **NAT**
  - **IPv6**