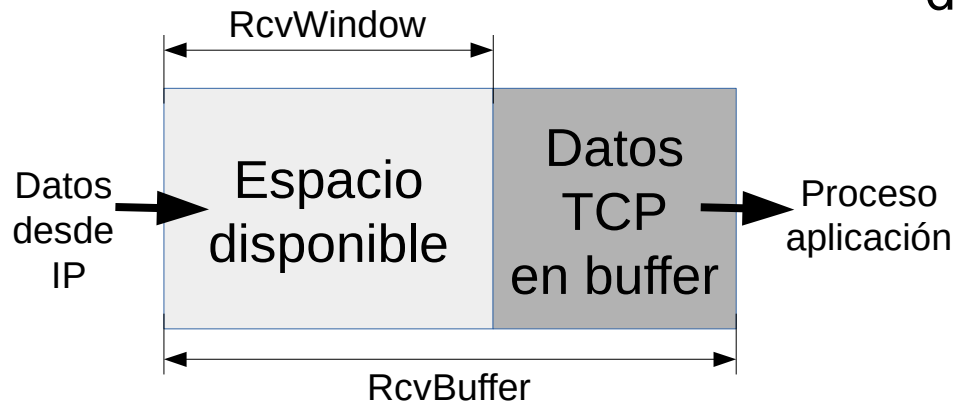


Capítulo 3: Continuación

- 3.1 Servicios de la capa de Transporte
- 3.2 Multiplexing y demultiplexing
- 3.3 Transporte sin conexión: UDP
- 3.4 Principios de transferencia confiable de datos
- 3.5 Transporte orientado a la conexión: TCP
 - Estructura de un segmento
 - Transferencia confiable de datos
 - Control de flujo
 - Administración de conexión
- 3.6 Principios del control de congestión
- 3.7 Control de congestión en TCP

Control de flujo en TCP

- El lado receptor de TCP tiene un buffer receptor



- El proceso aplicación puede ser lento en la lectura desde la capa de transporte

- La idea es hacer coincidir la tasa de transmisión con la tasa de lectura de la aplicación

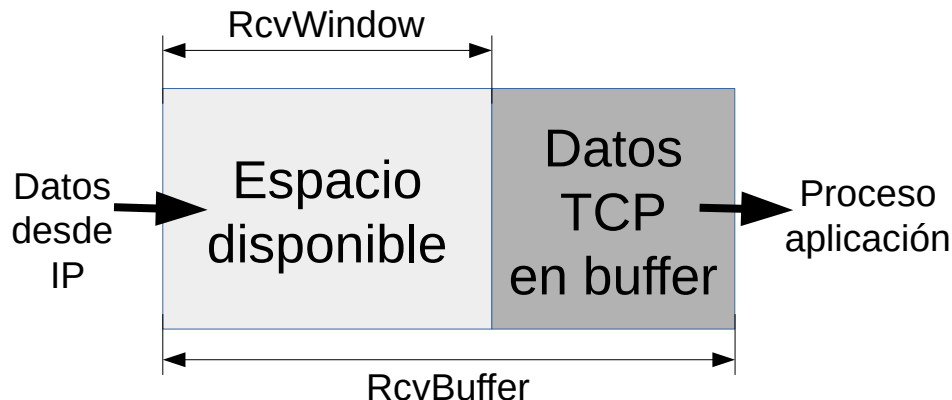
Control de flujo

El emisor no sobrecargará el buffer del receptor por transmitir demasiado rápido

Control de flujo en TCP

- Supongamos que el receptor descarta los segmentos fuera de orden
- Espacio libre en buffer

$$\text{RcvWindow} = \text{RcvBuffer} - (\text{LastByteRcvd} - \text{LastByteRead})$$



- El receptor comunica el espacio libre a través del valor de **RcvWindow** en los segmentos
- Así, el receptor limita los datos en tránsito (sin ACK) a **RcvWindow**
 - El emisor debe respetar el no envío de más datos que **RcvWindow**
 - Esto garantiza que el buffer del receptor no desborde (*overflow*)

Capítulo 3: Continuación

- 3.1 Servicios de la capa de Transporte
- 3.2 Multiplexing y demultiplexing
- 3.3 Transporte sin conexión: UDP
- 3.4 Principios de transferencia confiable de datos
- 3.5 Transporte orientado a la conexión: TCP
 - Estructura de un segmento
 - Transferencia confiable de datos
 - Control de flujo
 - Administración de conexión
- 3.6 Principios del control de congestión
- 3.7 Control de congestión en TCP

Administración de Conexión en TCP

- Transmisor y receptor TCP establecen una **conexión** antes de intercambiar segmentos de datos
 - TCP inicializa variables
 - Número de secuencia
 - Buffers
 - Información de control de flujo (e.g. RcvWindow)
 - Cliente: inicio de conexión

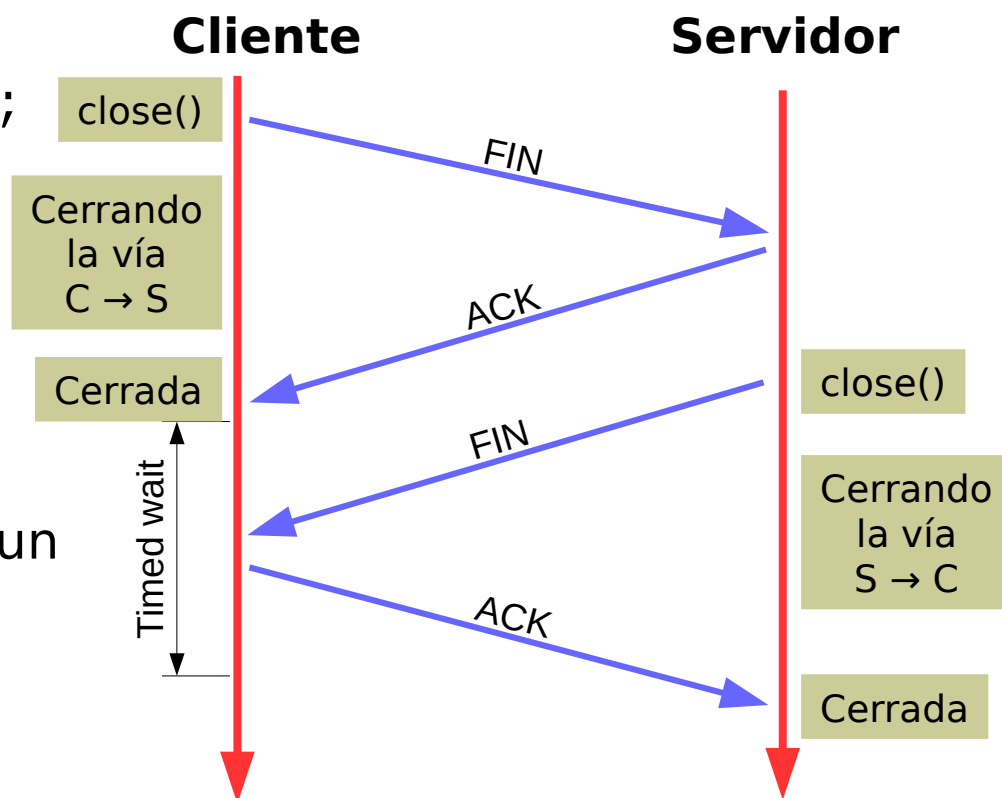
```
Socket clientSocket = new Socket("hostname", "port number");
```
 - Server: contactado por cliente

```
Socket connectionSocket = welcomeSocket.accept();
```
- **Apretón de manos de tres vías (three way handshake)**
 - Paso 1: cliente envía segmento TCP SYN al servidor
 - Especifica número de secuencia inicial cliente → servidor
 - no transporta datos
 - Paso 2: servidor recibe SYN
 - Responde con segmento SYN & ACK
 - Servidor prepara buffers
 - Especifica número de secuencia inicial servidor → cliente
 - Paso 3: cliente recibe SYN & ACK
 - Responde con segmento ACK, el cual ya puede contener datos

Administración de conexión en TCP

Cierre de conexión

- Cliente cierra socket
`clientSocket.close();`
- **Paso 1**
 - Cliente envía segmento TCP FIN al servidor
- **Paso 2**
 - Servidor recibe FIN, responde con ACK ante un cierre de conexión de la aplicación y envía FIN.



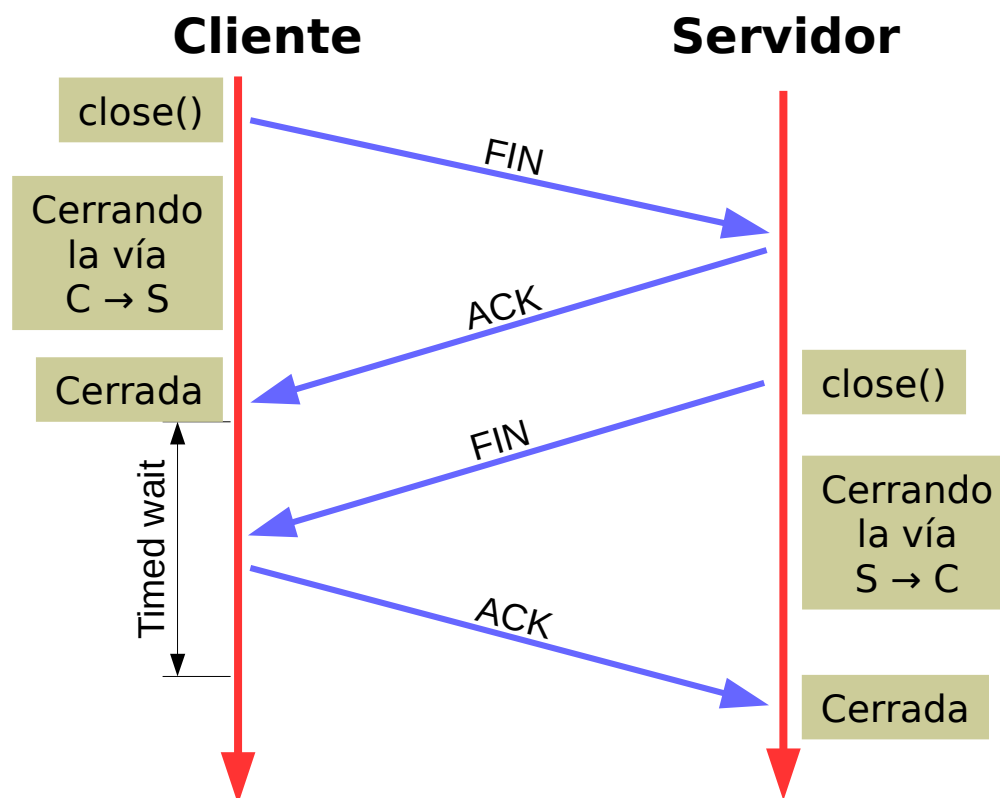
Administración de conexión en TCP

■ Paso 3

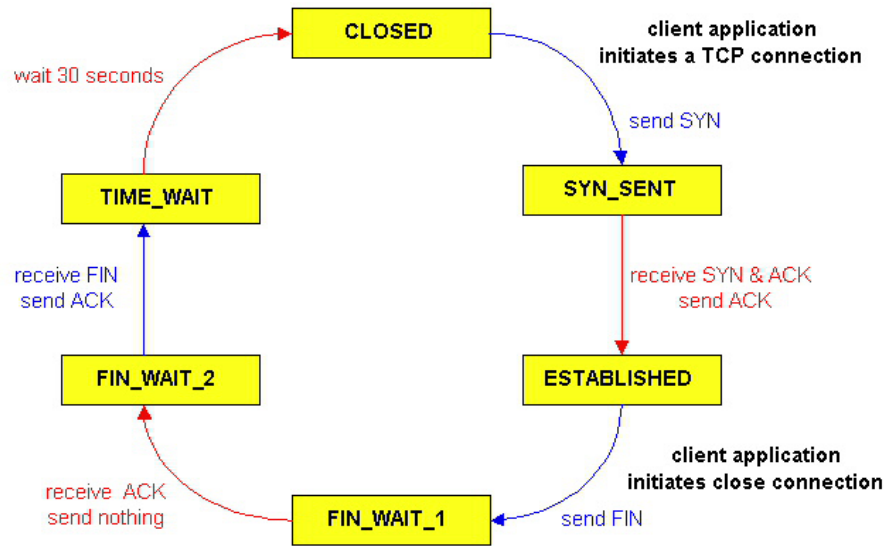
- Cliente recibe FIN, responde con ACK
- Entra en **timed wait**
- Responderá con ACK a FINs recibidos

■ Paso 4

- Servidor recibe ACK
- Pasa a conexión cerrada

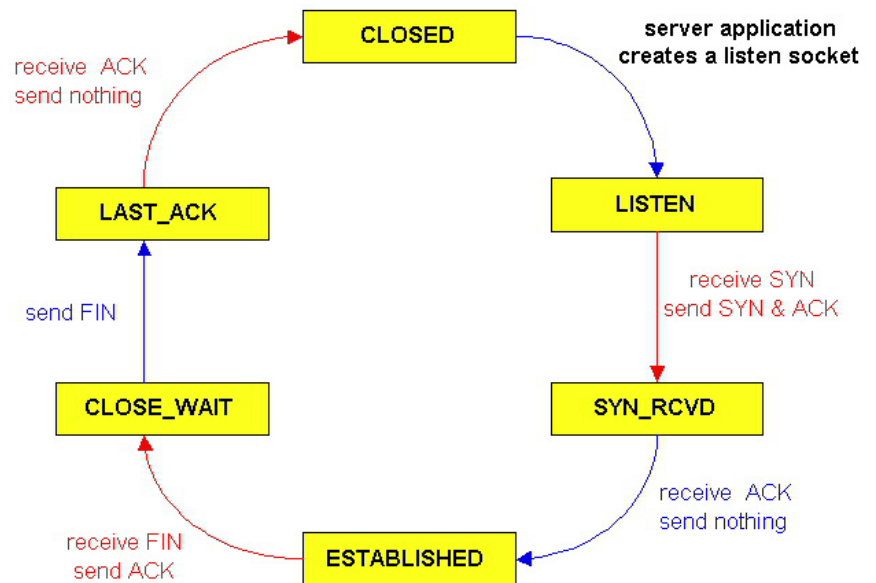


Administración de la Conexión TCP

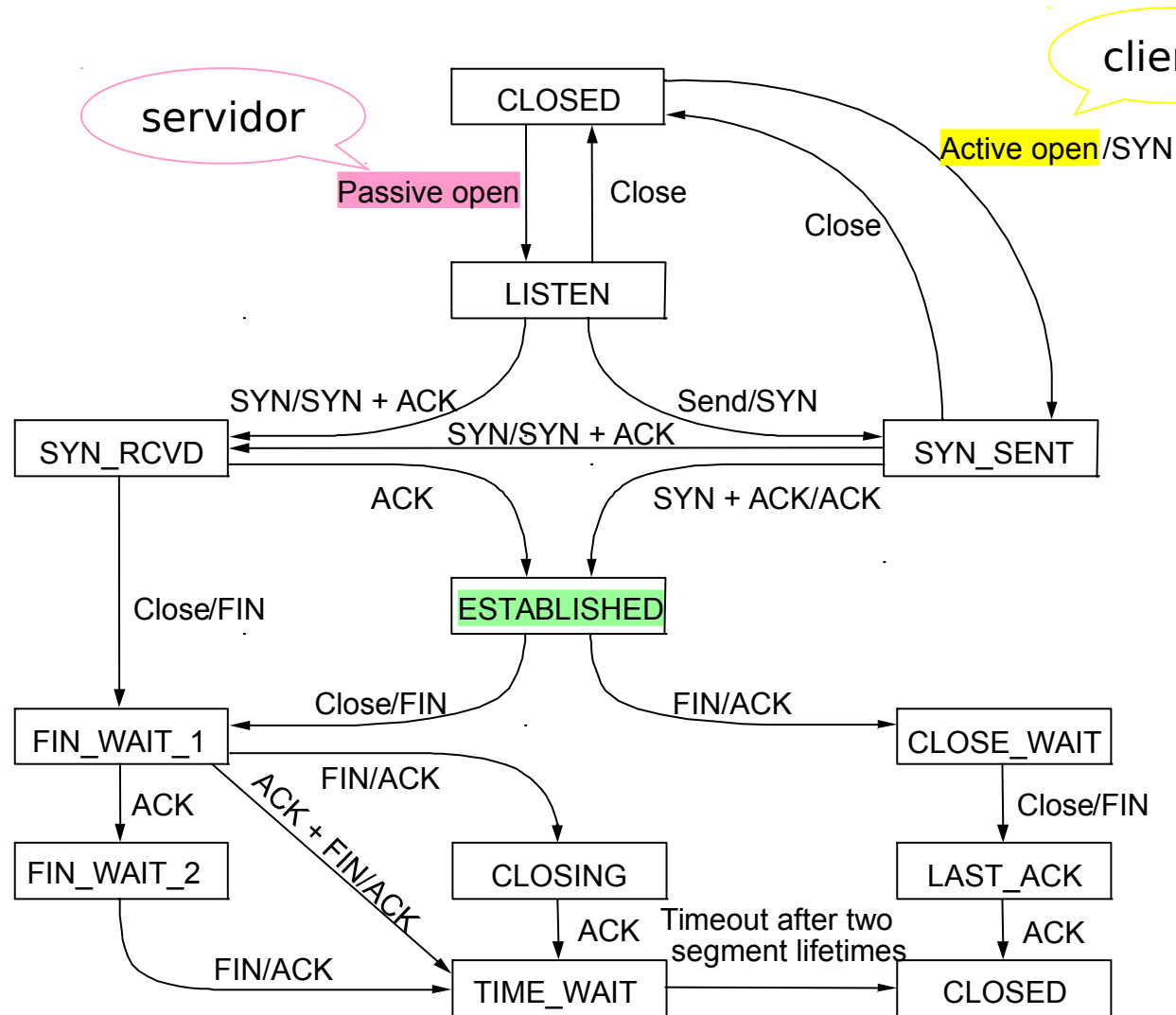


Ciclo de vida del cliente TCP

Ciclo de vida del servidor TCP



Administración de la Conexión TCP



Capítulo 3: Continuación

- 3.1 Servicios de la capa transporte
- 3.2 Multiplexing y demultiplexing
- 3.3 Transporte sin conexión: UDP
- 3.4 Principios de transferencia confiable de datos
- 3.5 Transporte orientado a la conexión: TCP
 - Estructura de un segmento
 - Transferencia confiable de datos
 - Control de flujo
 - Administración de conexión
- 3.6 Principios del control de congestión
- 3.7 Control de congestión en TCP

Principios del control de congestión

Congestión

- Informalmente:
 - Demasiadas fuentes envían demasiados datos demasiado rápido para que la red lo maneje
 - ¡Es distinto a control de flujo!
- Manifestaciones:
 - Pérdidas de paquetes (buffer overflow en routers)
 - Grandes retardos (colas en los routers)
- ¡Uno de los problemas top-10!

Estrategias de control de congestión

Control de congestión asistido por la red

- Los routers proveen realimentación a los sistemas extremos
 - Bit único indicando congestión (SNA, DECbit, TCP/IP ECN, ATM)
 - La red informa explícitamente al emisor la tasa de datos que el router puede soportar

Control de congestión extremo a extremo

- No hay realimentación explícita de la red
- La congestión es inferida a partir de las pérdidas y retardos observados por los sistemas extremos
- Es la estrategia usada por TCP

Control de congestión en ATM

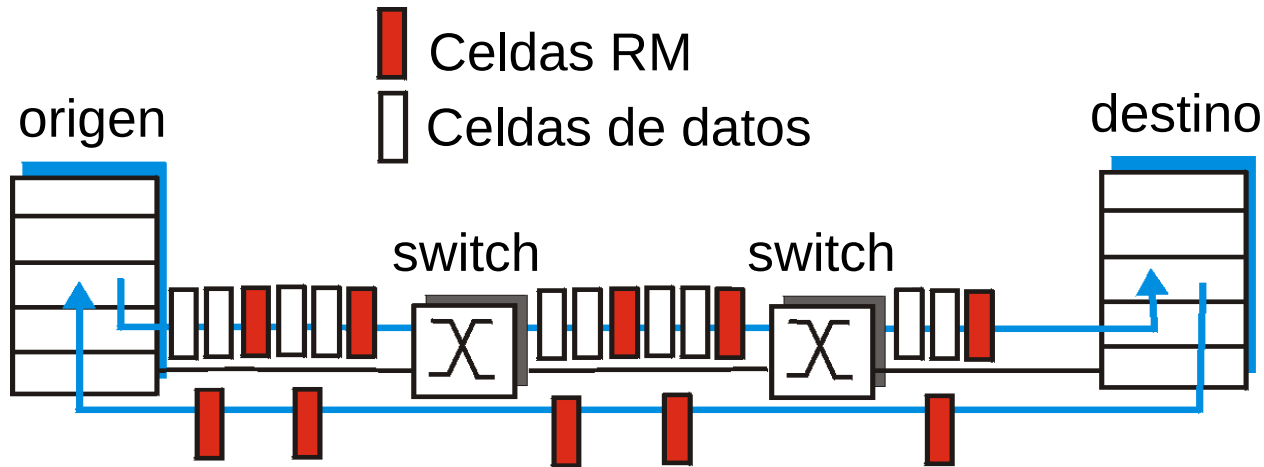
- **ABR**

- Available bit rate
- Modo de servicio elástico
- Si el camino del Tx no tiene “mayor carga”:
 - Tx debería usar ancho de banda disponible
- Si el camino del Tx está congestionado:
 - Tx reduce a un mínimo la tasa garantizada

- Celdas **RM**

- Resource management
- Enviadas a intervalos por Tx y entre celdas de datos
- Bits en celda RM fijados por switches (“asistido por la red”)
 - Bit **NI**: no incrementar tasa (congestión moderada)
 - Bit **CI**: Congestion Indication
- Las celdas RM son devueltas al Tx por el Rx con los bits intactos

Control de congestión en ATM ABR



- Campo **ER** (explicit rate) de dos bytes en celda RM
 - Un switch congestionado puede disminuir el valor de ER en la celda
 - La tasa de envío del emisor se ajusta a la tasa mínima soportable en todo el camino
- Bit **EFCI** en celdas de datos
 - Es fijado a 1 en el switch congestionado
 - Si la celda de datos precedente tiene el EFCI marcado, el emisor activa el bit **CI** en la celda RM devuelta

Capítulo 3: Continuación

- 3.1 Servicios de la capa de Transporte
- 3.2 Multiplexing y demultiplexing
- 3.3 Transporte sin conexión: UDP
- 3.4 Principios de transferencia confiable de datos
- 3.5 Transporte orientado a la conexión: TCP
 - Estructura de un segmento
 - Transferencia confiable de datos
 - Control de flujo
 - Administración de conexión
- 3.6 Principios del control de congestión
- 3.7 Control de congestión en TCP