Planificación y Administración de Redes: El nivel de Transporte

IES Gonzalo Nazareno
CONSEJERÍA DE EDUCACIÓN

Jesús Moreno León
Raúl Ruiz Padilla

j.morenol@gmail.com

Septiembre 2010

Estas diapositivas son una obra derivada de las transparencias del Grupo de Sistemas y Comunicaciones de la Universidad Rey Juan Carlos
Puede encontrarse una versión de este documento en
http://gsyc.es/moodle

Introducción

El nivel de transporte gobierna el acceso múltiple a la red de los distintos procesos de la misma máquina que quieran usarla: en TCP/IP esto se consigue a través de los puertos.

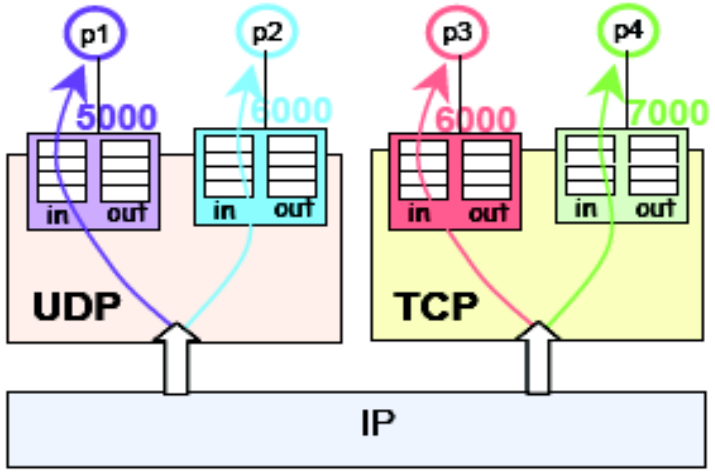
Hay dos protocolos principales que ofrecen nivel de transporte:

- UDP: no orientado a conexión y no fiable
- TCP: orientado a conexión y fiable

Puertos

El nivel de transporte debe preocuparse de saber a qué proceso va destinada la información que le llega del nivel de red. Por ello, los procesos que usan la red lo hacen a través de puertos.

Cada puerto en el nivel de transporte proporciona a una aplicación un punto de acceso a la red, de forma que pueda dialogar con otro proceso de una máquina remota.



Puertos

Los puertos se identifican por un número de 16 bits

Los puertos TCP y UDP se manejan por separado; el puerto 7 UDP y el puerto 7 TCP son distintos

Los puertos menores de 1024 (puertos privilegiados) están asignados y reservados universalmente a aplicaciones de red conocidas

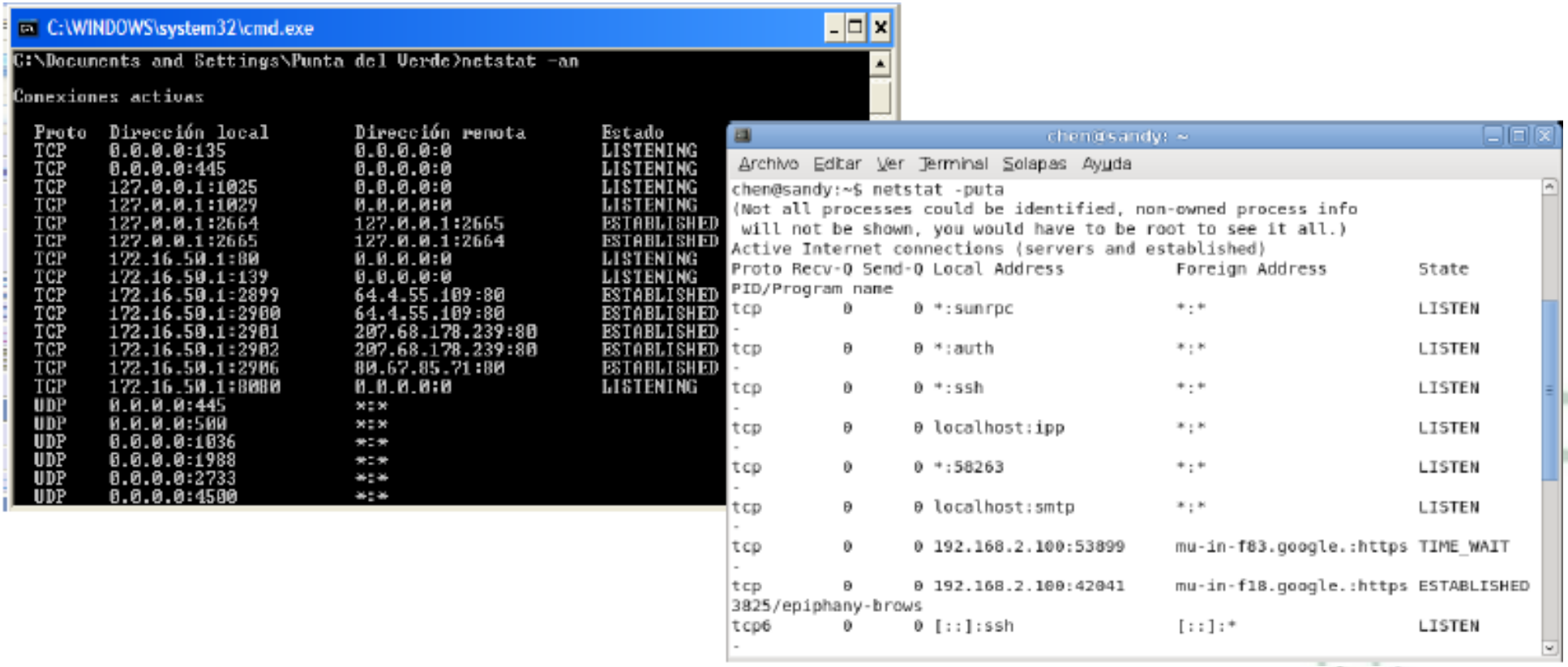
Puertos

Un servidor www es un proceso esperando peticiones en el puerto 80 de una máquina

Un navegador, desde otra máquina, hará peticiones al puerto 80 del servidor y escuchará las respuestas en un puerto suyo no privilegiado

Puertos

Netstat (Network Statistics) es una herramienta de línea de comandos que muestra un listado de las conexiones activas de un ordenador, tanto entrantes como salientes.

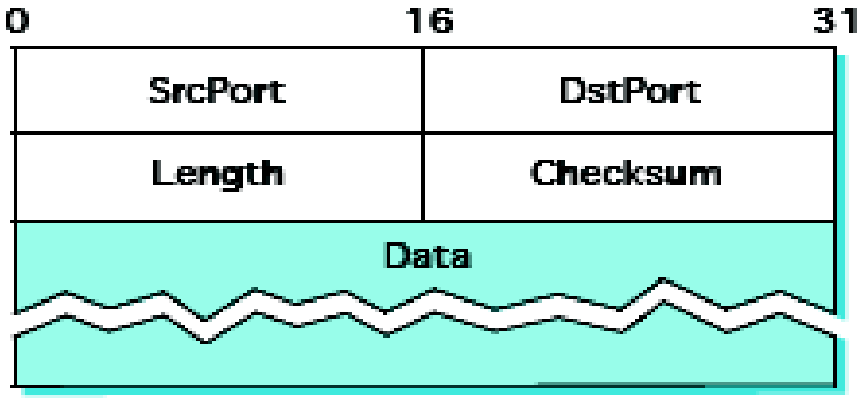


UDP: User Datagram Protocol

UDP es un protocolo sencillo que implementa un nivel de transporte no orientado a conexión y no fiable.

Ofrece un servicio de entrega de datagramas no ordenado, no fiable y que no proporciona control de flujo

Los datagramas UDP se encapsulan dentro de la parte de datos de un datagrama IP



UDP: User Datagram Protocol

El servicio ofrecido por UDP sólo aumenta el ofrecido por IP en:

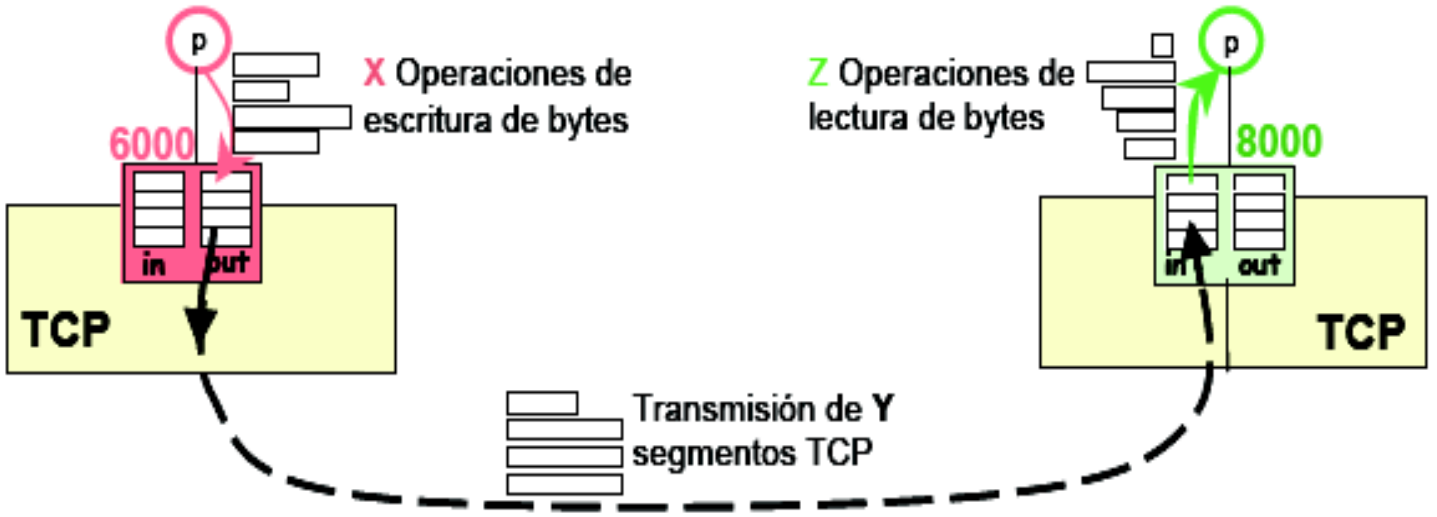
- Número de puerto
- Checksum opcional

Por ello el servicio es no fiable (pueden perderse, duplicarse o desordenarse datagramas)

Es un protocolo mucho más ligero que TCP, por lo que para aplicaciones que se ejecuten dentro de una subred (no tienen que atravesar encaminadores, por lo que las pérdidas son más improbables) puede compensar

TCP: Transmission Control Protocol

- **Orientado a conexión**: hay fase anterior y posterior al envío de datos
- Envío de datos **fiable**: sin pérdidas, duplicados o desorden
- Las conexiones son **full duplex**: ambos lados pueden enviar datos simultáneamente
- **Control de flujo**: mecanismo para evitar que el emisor inunde al receptor
- Envío de datos como **flujo de bytes**:



Servicio orientado a la conexión

La transmisión de datos en una conexión TCP presenta las siguientes fases:

- Establecimiento de la conexión
- Envío de datos
- Finalización de la conexión

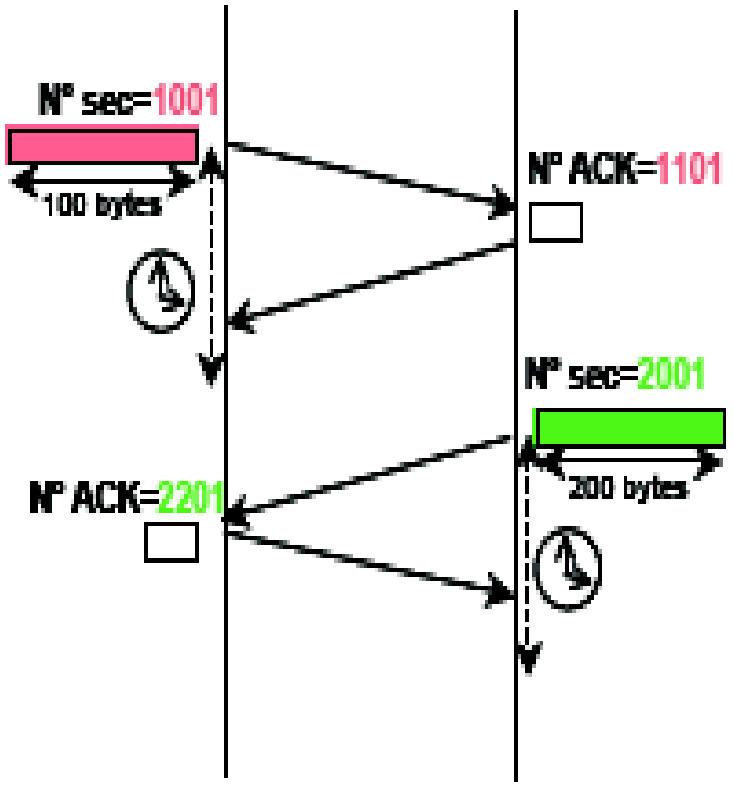
Ambos extremos pueden transmitir y recibir simultáneamente

Servicio fiable

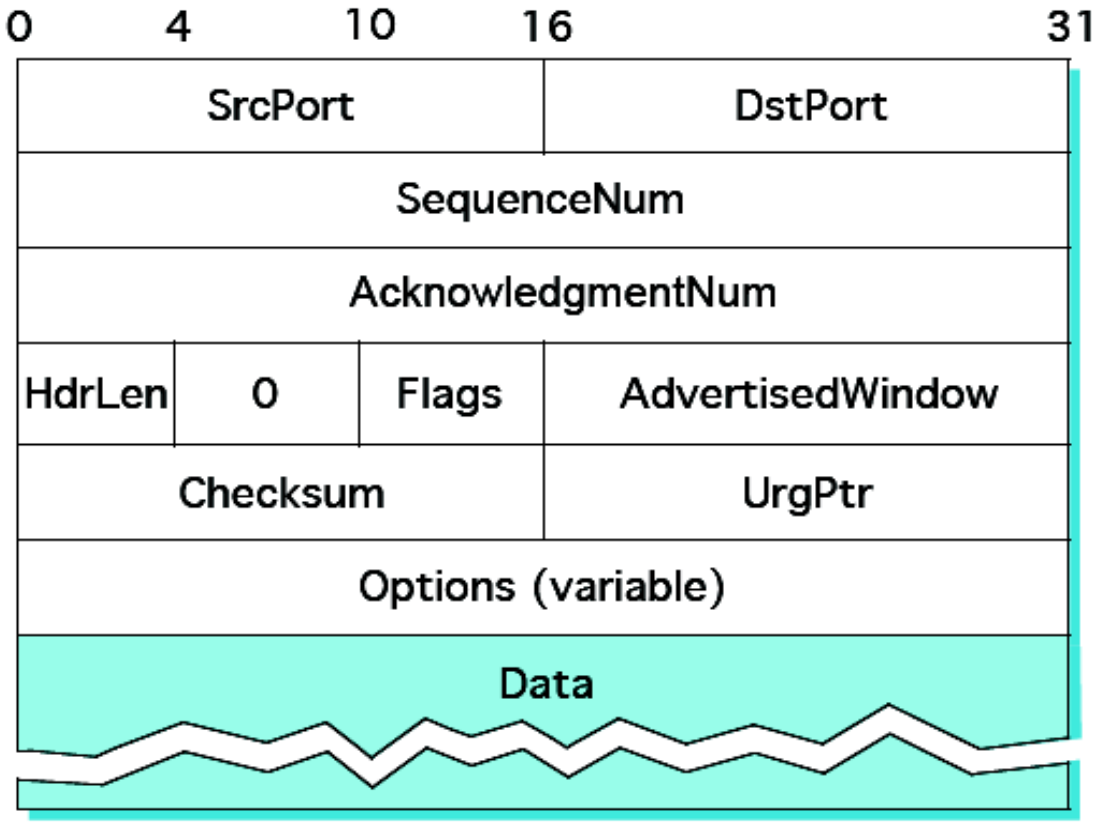
Éste es el primer nivel en TCP/IP en el que se proporciona fiabilidad. Su objetivo es recuperarse de las posibles pérdidas y desorden producido por IP

Idea básica:

- Los segmentos con datos llevan un número de secuencia
- El receptor de los datos debe mandar asentimientos (ACK)
- TimeOut y protocolo de ventana
- El receptor reordena segmentos y descarta duplicados



Formato de segmento



Número de secuencia

Cada segmento con datos lleva un número de secuencia `SequenceNum` de 32 bits

El número de secuencia numera bytes y no segmentos

Al establecerse la comunicación se elige un número de secuencia inicial

Número de asentimiento (ACK)

El receptor de segmentos de datos tiene que asentir los que le llegan correctamente, activando el flag `ACK` y rellenando el campo `AcknowledgementNum`

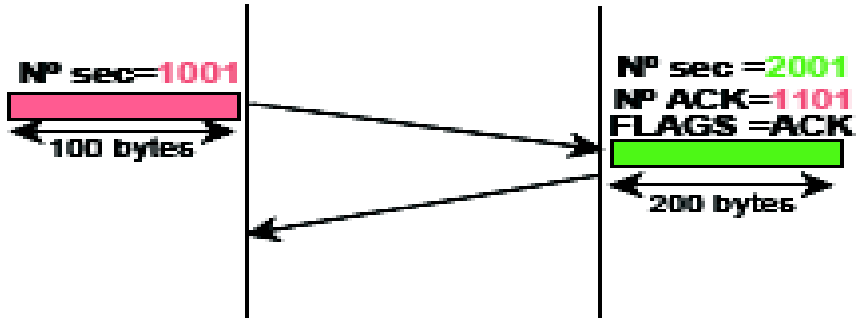
No es necesario enviar un asentimiento por cada segmento con datos recibido

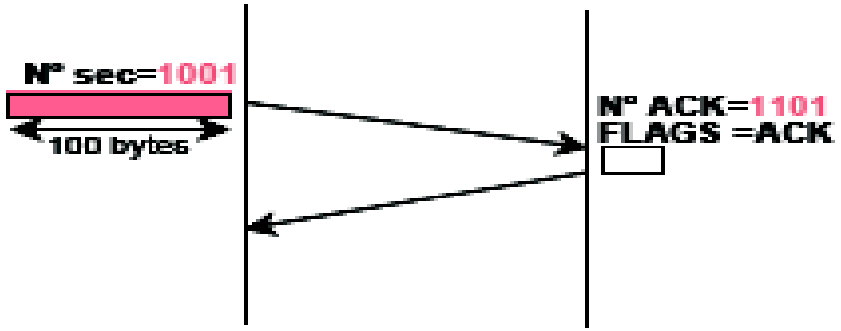
El número de asentimiento indica el número de secuencia que se espera recibir, asintiendo de esta forma hasta el byte anterior incluido

No hay rechazo selectivo

Si hay datos que enviar, se aprovecha ese segmento para enviar un asentimiento *a recucas* (piggybacking)

Número de asentimiento (ACK)





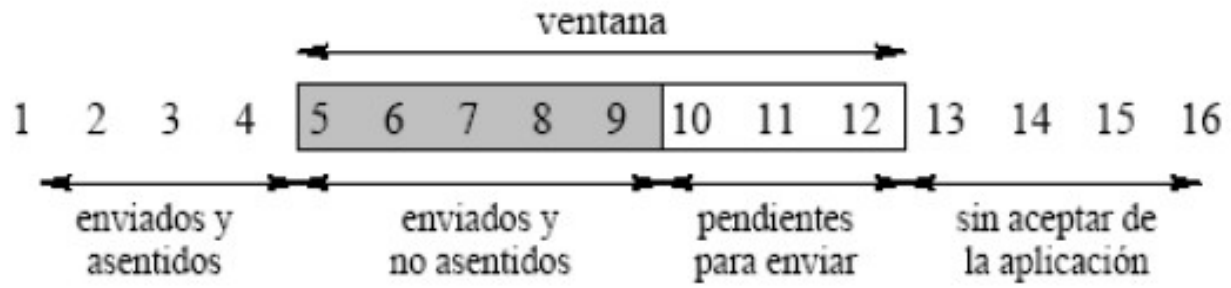
Cada lado de la conexión utiliza sus números de secuencia(partiendo de su número de secuencia inicial) y asiente los números de secuencia que está usando el otro extremo

Ventana anunciada (o ventana de flujo)

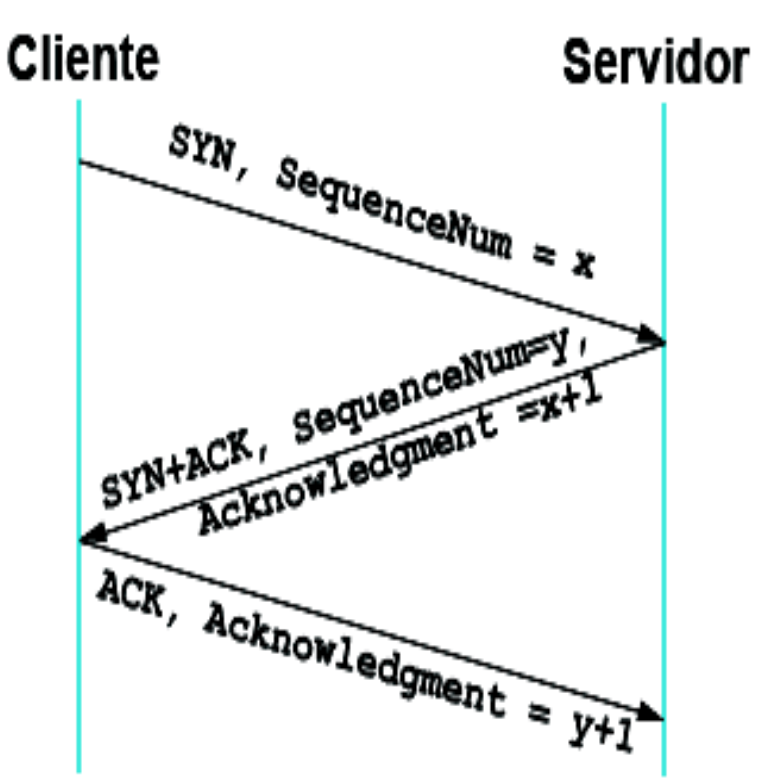
Se usa un protocolo de ventana anunciada para coordinar el envío de segmentos de datos

El receptor indica en el campo `AdvertisedWindow` el número de bytes (a partir del indicado en el número de asentimiento) que está dispuesto a recibir del emisor

El emisor puede transmitir esos bytes aunque no reciba asentimientos, pero una vez enviados tendrá que parar hasta recibir un nuevo asentimiento (con un nuevo valor de ventana)



Establecimiento y cierre de la conexión



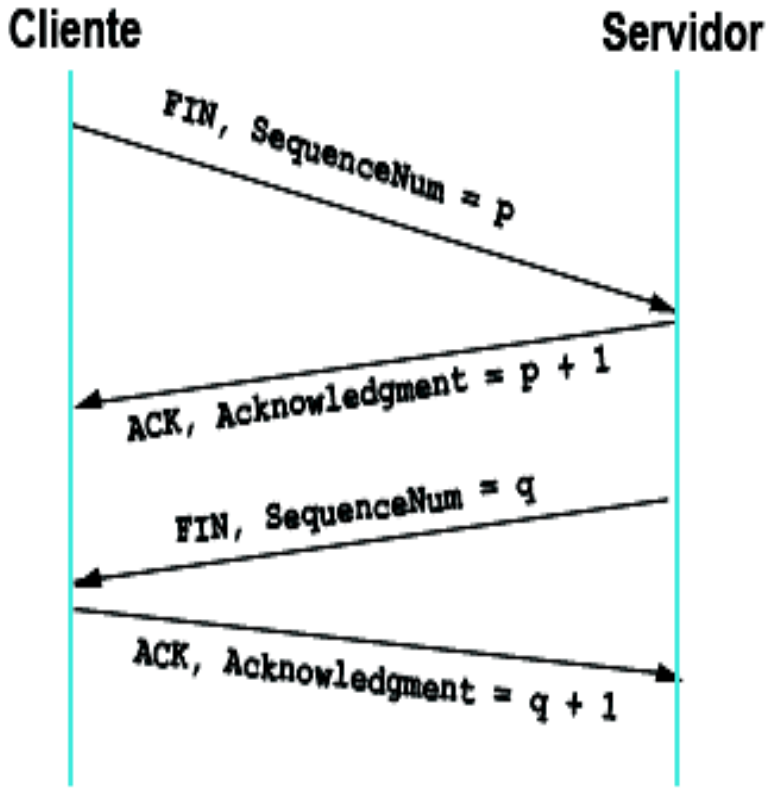


Diagrama de estados en el establecimiento de la conexión

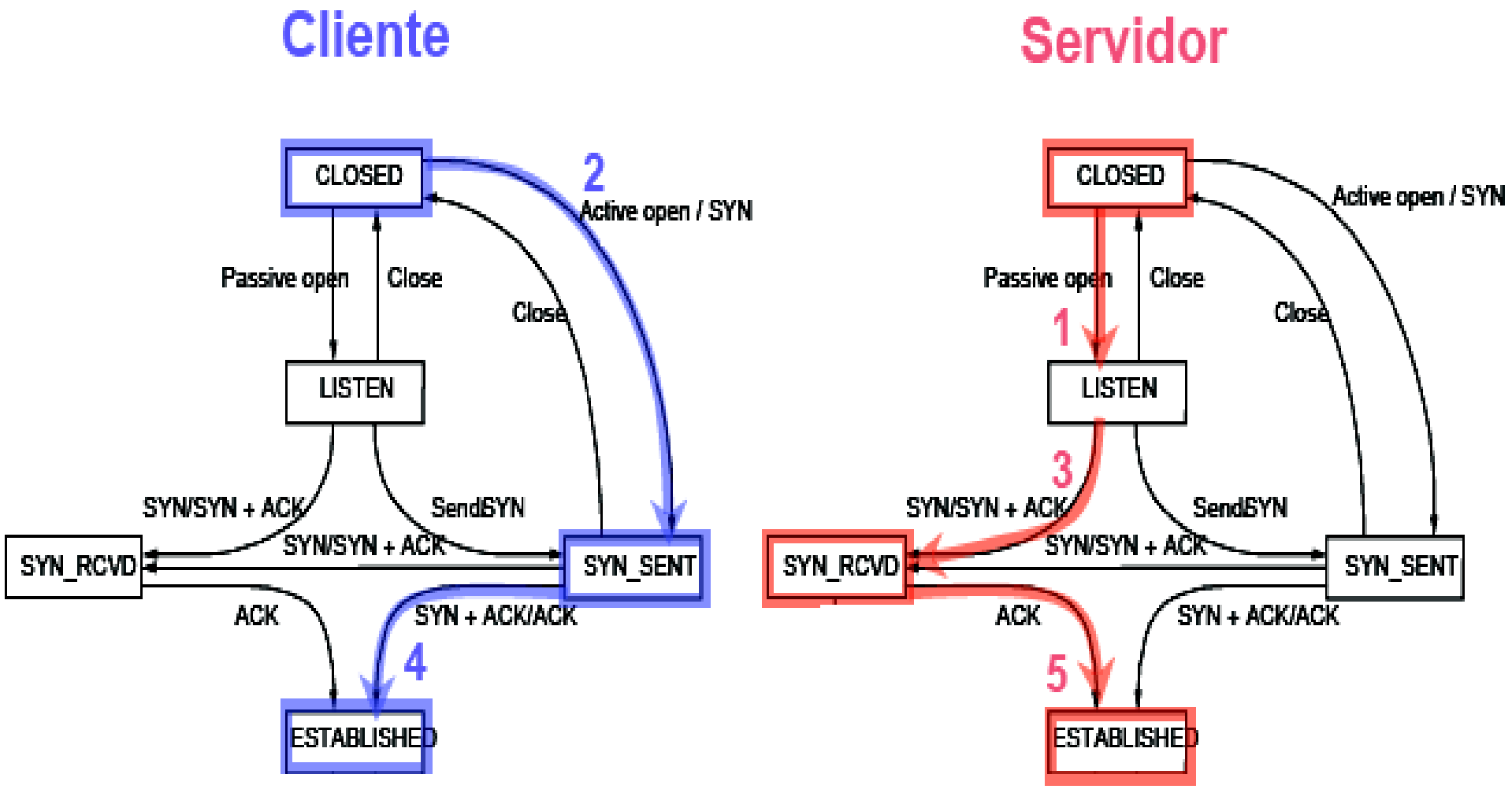
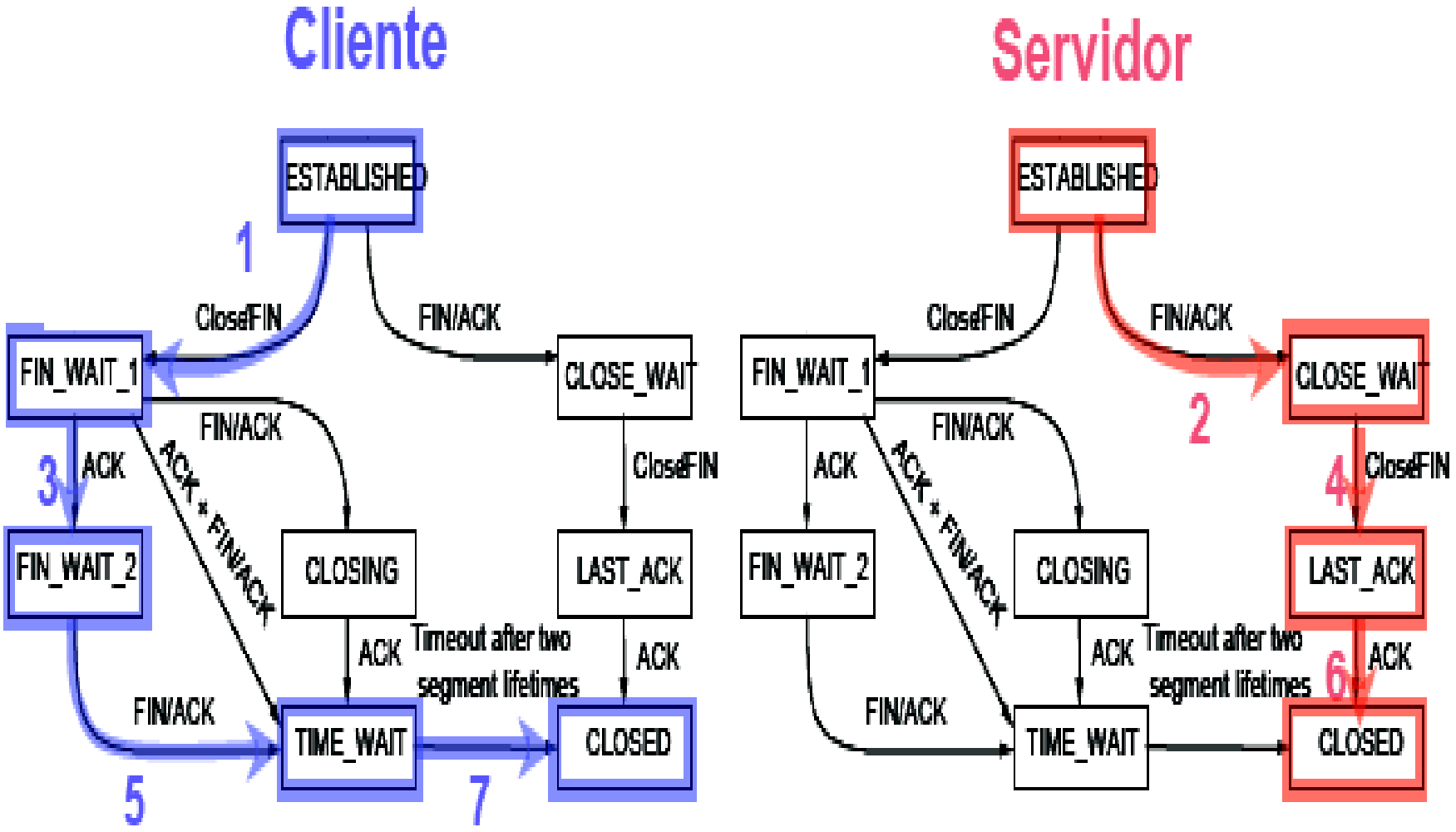
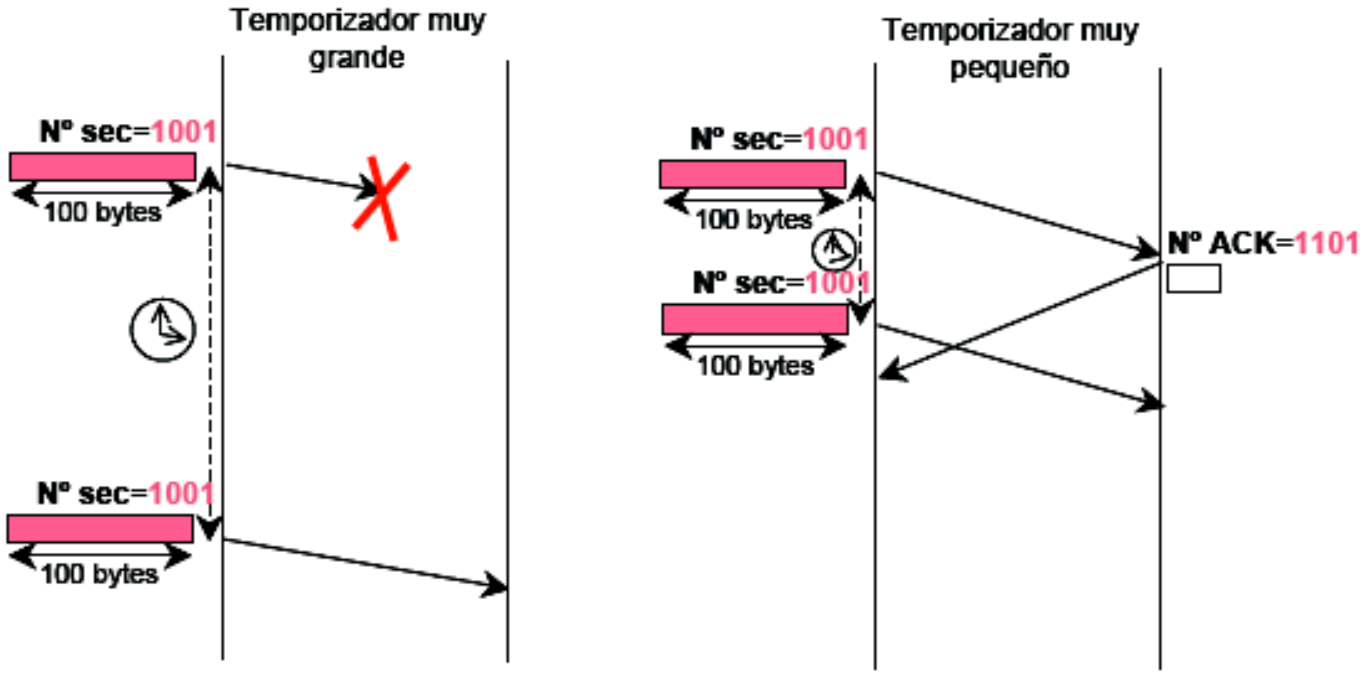


Diagrama de estados en el cierre de la conexión



Plazos para asentimiento

Cuando se envía un segmento se arranca un temporizador para esperar su asentimiento. Transcurrido el plazo marcado en el temporizador (timeout), si no se ha recibido el ACK, se retransmite. ¿Qué plazo ponemos?



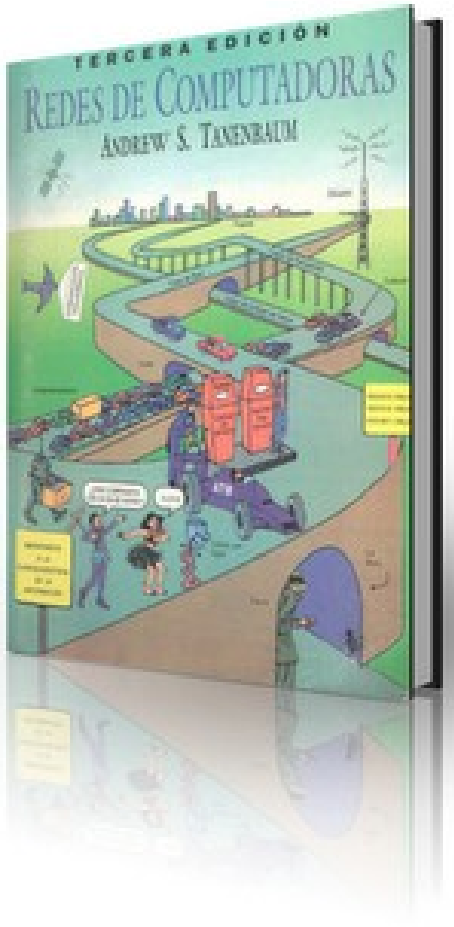
Plazos para asentimiento

Se utiliza un algoritmo adaptativo, que se adapta a lo que ocurre en la conexión

Para cada segmento se calcula el Tiempo de Ronda (Round Trip Time → RTT): tiempo entre que se envía el segmento y se recibe su asentimiento

Existen diferentes algoritmos, pero el original usa un plazo de 2 veces el RTT

Bibliografía



A. Tanembaum, Redes de Computadores (4a ed.): Capítulo 6

Apartados: 6.5