



# **Protocolos de Transporte**

**Ing. Norberto Gaspar Cena**  
Redes de Información

4to Año Ingeniería en Sistemas de Información

# La capa de Transporte y sus Servicios

Proporciona una comunicación lógica entre procesos de distintos hosts

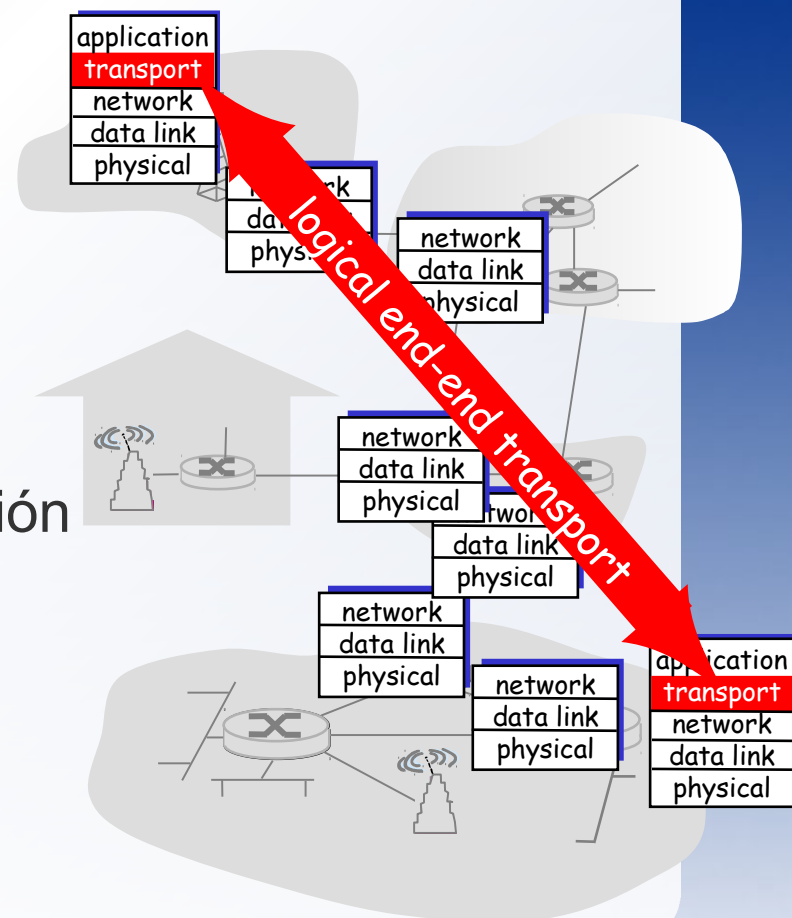
Esta implementado en los sistemas terminales (segmentos)

Amplía el servicio de entrega IP

- Entrega Proceso a Proceso
- Multiplexación y Demultiplexación

Dos Protocolos de transporte

- UDP y TCP



# Multiplexación y Demultiplexación

La capa de transporte recibe segmentos procedentes de la capa de red

Tiene la responsabilidad de entregar los datos al proceso de la aplicación apropiada

- FTP, HTTP, TELNET

Puertos bien conocidos ([www.iana.org](http://www.iana.org) [RFC 3232])

Los procesos tienen **sockets** para la comunicación entre la capa de red y aplicación

- Identificador único
- Host de origen multiplexación
- Host de destino demultiplexación
- Necesario en todas las comunicaciones

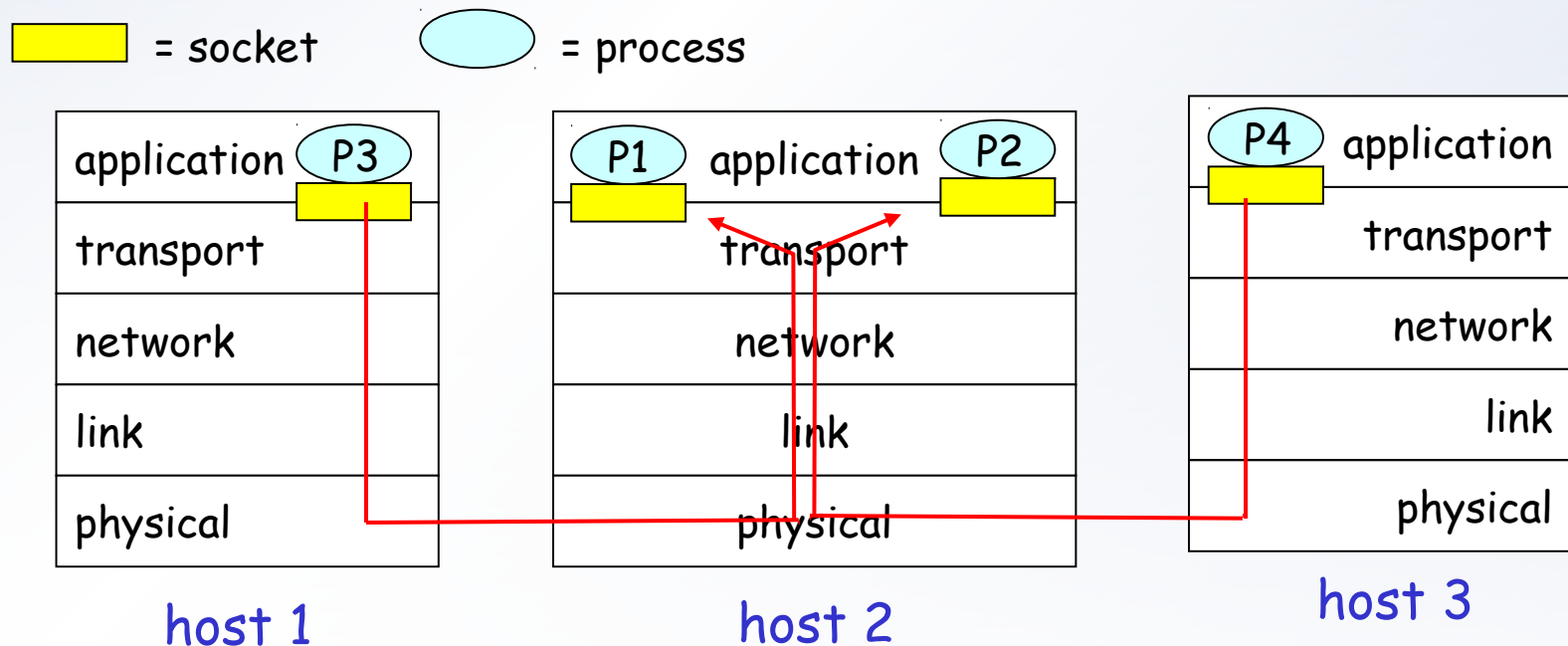
# Multiplexación y Demultiplexación

## No orientado a la conexión

- Queda identificado por ip destino/puerto destino

## Orientado a la conexión

- Queda identificado por ip origen/puerto origen/ip destino/puerto destino



# Transporte No Orientado a la Conexión

User Datagram Protocol (UDP) [RFC 768]

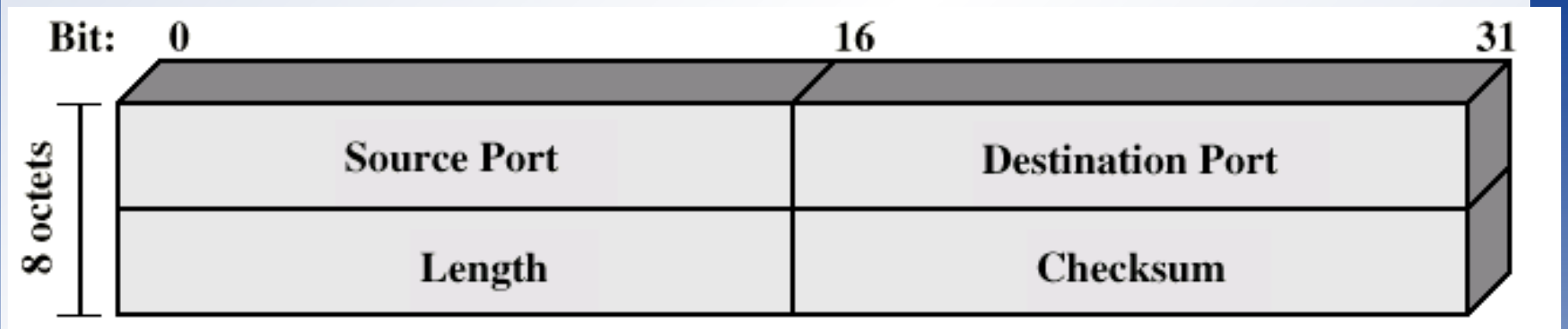
Hace lo mínimo que un protocolo de transporte tiene que hacer

- Multiplexación, Demultiplexación y Control de errores

Ventajas

- Mejor control en el nivel de la aplicación sobre de qué datos se envían y cuándo
- Sin establecimiento de la conexión
- Sin confirmación de estado de la conexión
- Poca sobrecarga debida a la cabecera de los paquetes

# Formato del Paquete UDP



# Transporte Orientado a la Conexión

Transport Control Protocol (TCP) [RFC 793] [RFC 1122] [RFC 1323] [RFC 2018] [RFC 2581]

Fiable y Orientado a la conexión

- Se debe “establecer una comunicación”
- Detección de errores, control de flujo, temporizadores, etc

Proporciona un servicio full-duplex

Establecer una conexión TCP

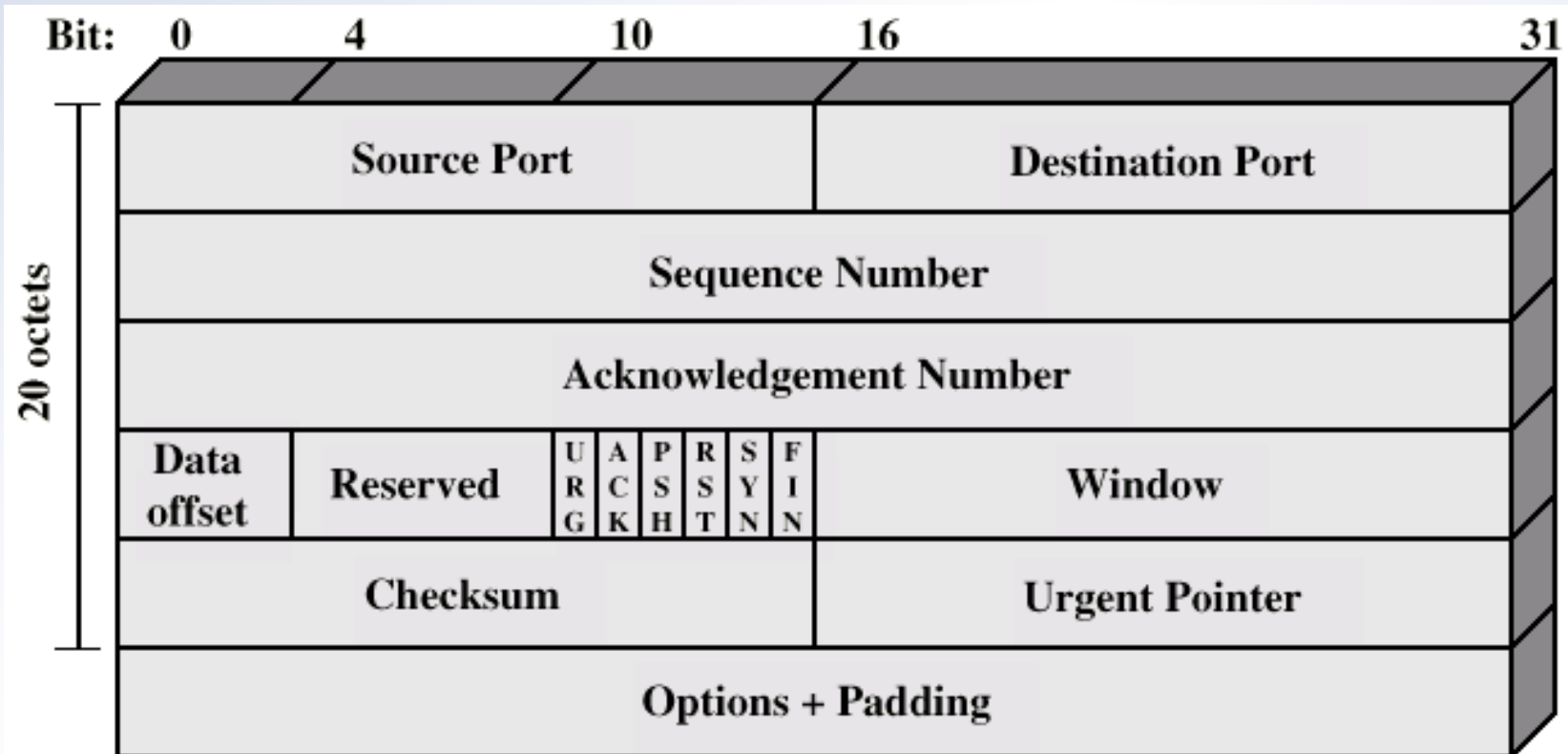
- Se intercambian 3 segmentos

Buffer de emisión

Utilizado por aplicaciones que requieran garantías de entrega



# Estructura del Segmento TCP





# Transporte Orientado a la Conexión

Establecimiento de la conexión

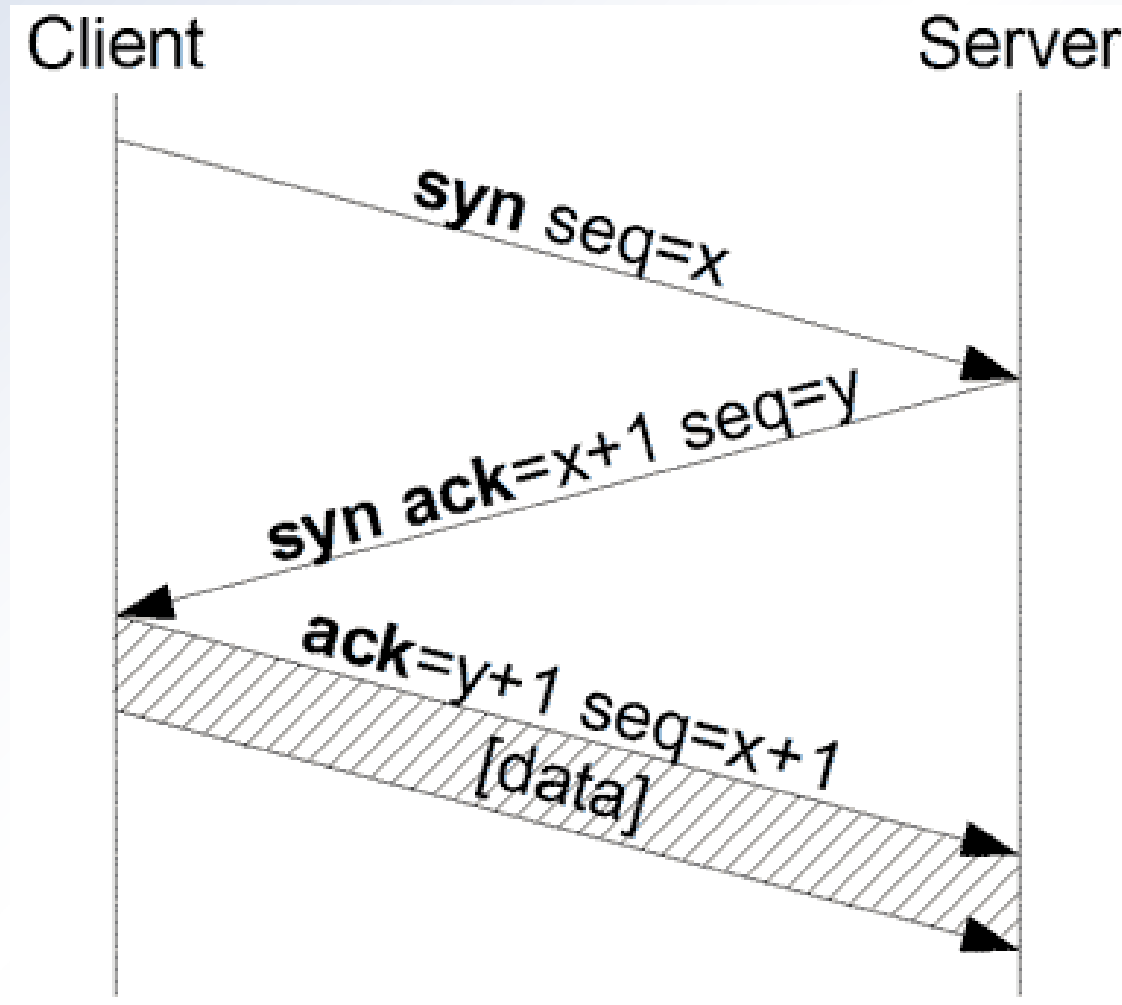
*Negociación de 3 pasos (3-way handshake, SYN, SYN/ACK, ACK)*

- *El cliente envía un paquete SYN*
  - El servidor puede responder con un paquete RST
- El servidor responde con un paquete SYN/ACK
- El cliente responde con un paquete ACK

Apertura pasiva mediante socket (Servidor)

Números de secuencias de 32bits

# Transporte Orientado a la Conexión



# Transporte Orientado a la Conexión

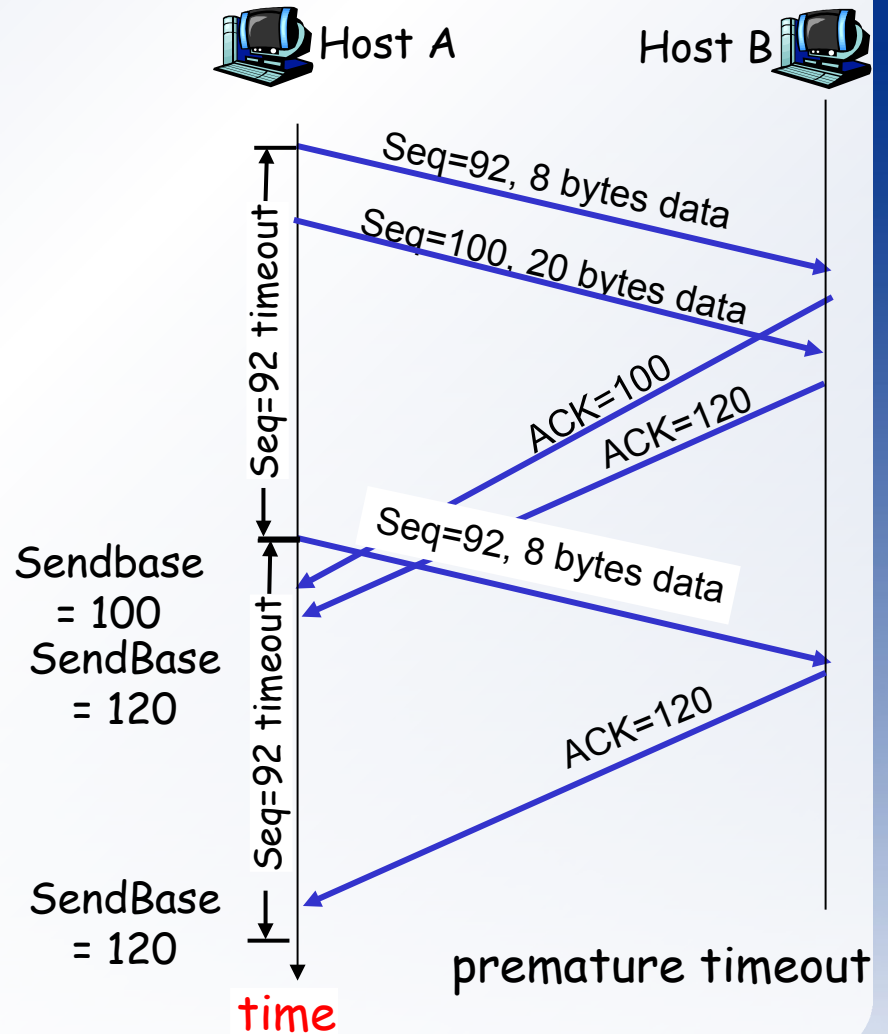
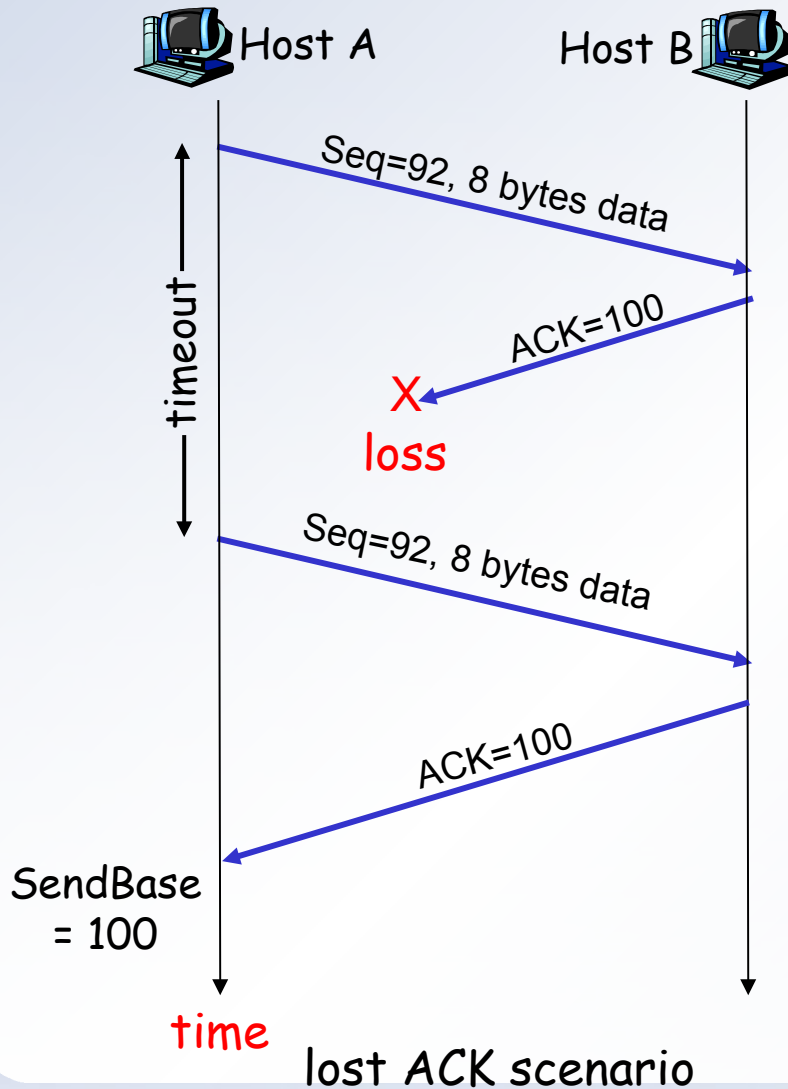
Con el “circuito virtual” creado se comienza con el envío de datos

TCP consume mas recursos (CPU y Ancho de Banda)

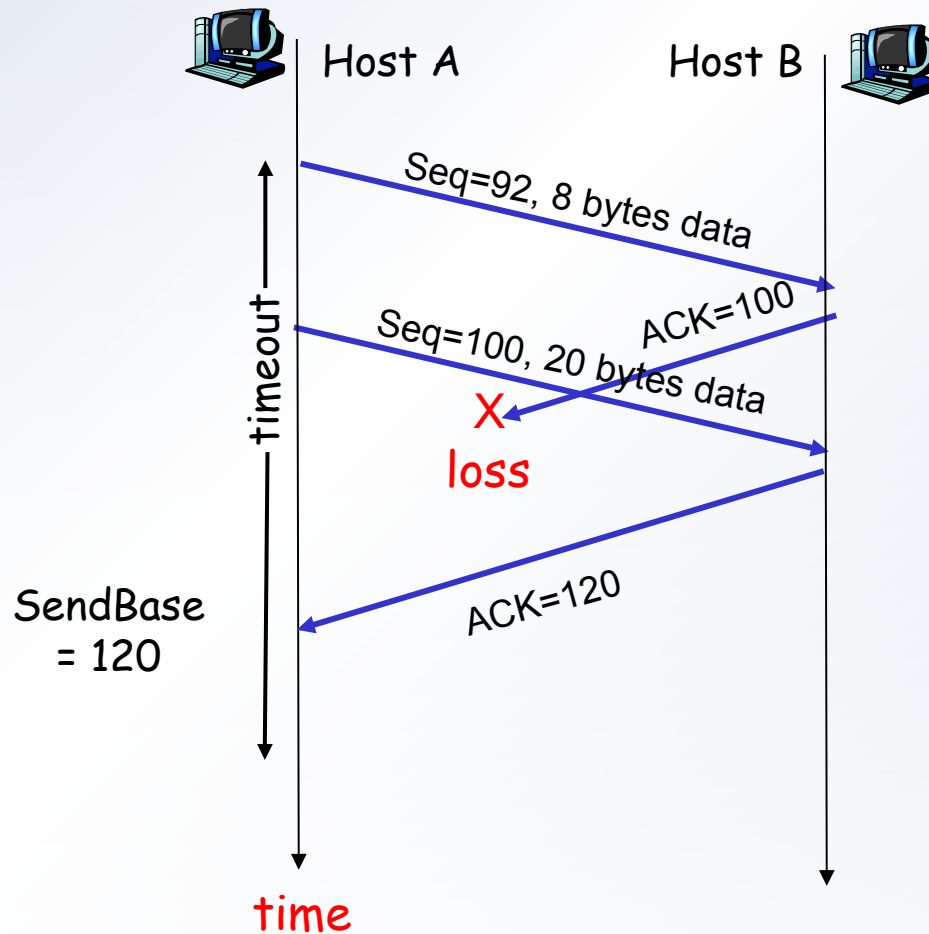
Implementa Ventana deslizante (Time-out y Retransmisión)

Tamaño de la ventana en bytes

# Escenarios TCP



# Escenarios TCP



Cumulative ACK scenario

# TCP – Control de Flujo

TCP no tiene forma de saber cual es el máximo volumen de datos que puede transmitir a la red

Si la red se satura comenzará a eliminar descartar segmentos

- Deberán ser retransmitidos, congestionando más la red

La técnica consiste en comenzar enviando un volumen de datos pequeño

Este volumen va aumentando hasta llegar al umbral de saturación

Una vez llegado a la saturación se reduce la tasa de envío

# TCP – Control de Flujo

## Slow-Star

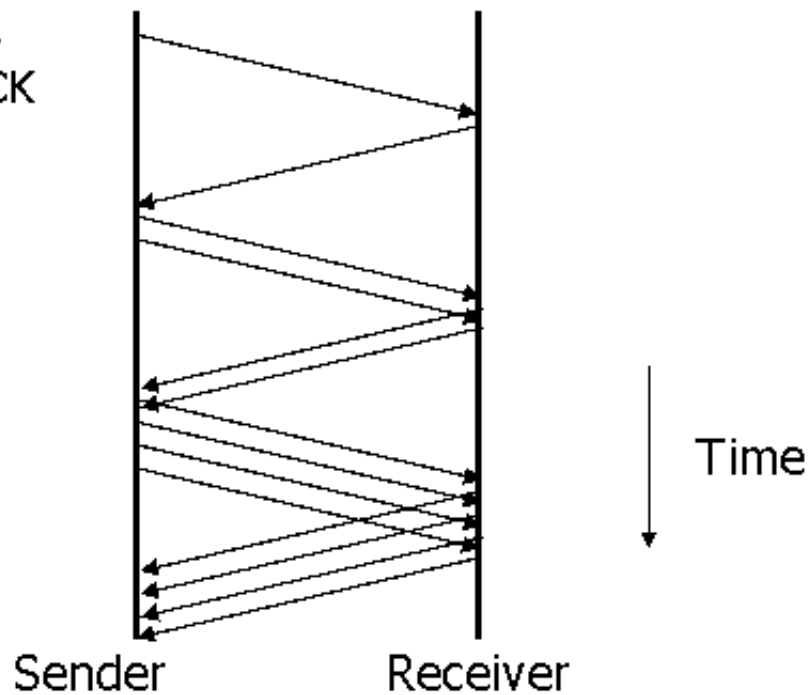
- La ventana de congestión se inicia con el valor de un segmento de tamaño máximo (MSS)
- Cada vez que se recibe un ACK, la ventana de congestión se incrementa en tantos bytes como hayan sido reconocidos en el ACK recibido.
  - En la práctica, esto supone que el tamaño de la ventana de congestión se doble por cada RTT (Round-Trip Time)
- Cuando un ACK no llega al transmisor:
  - Se toma como una señal de congestión en la red y se reinicia la ventana de congestión a un MSS.



# TCP – Control de Flujo

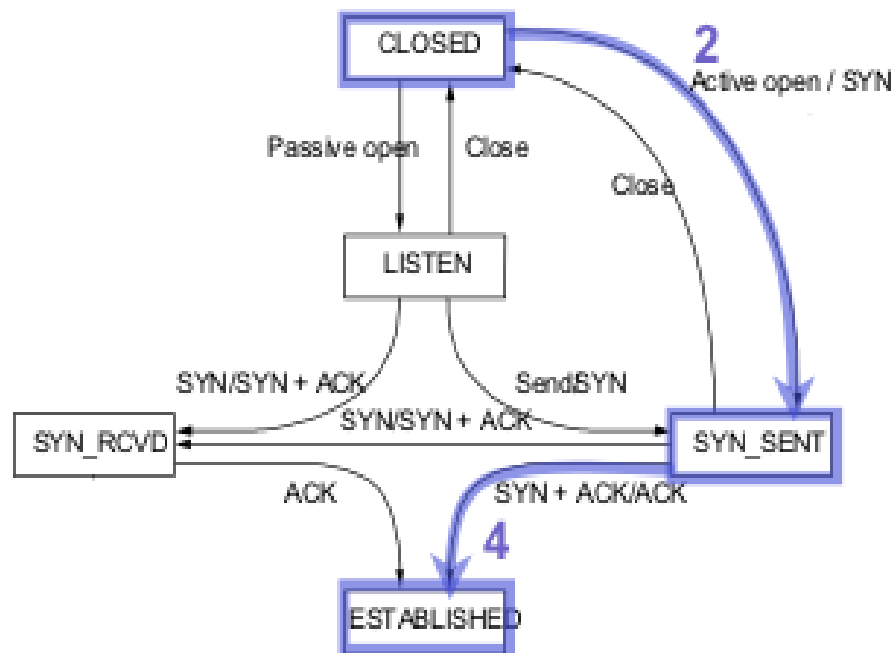
## TCP Slow Start

Increase by 1  
packet per ACK

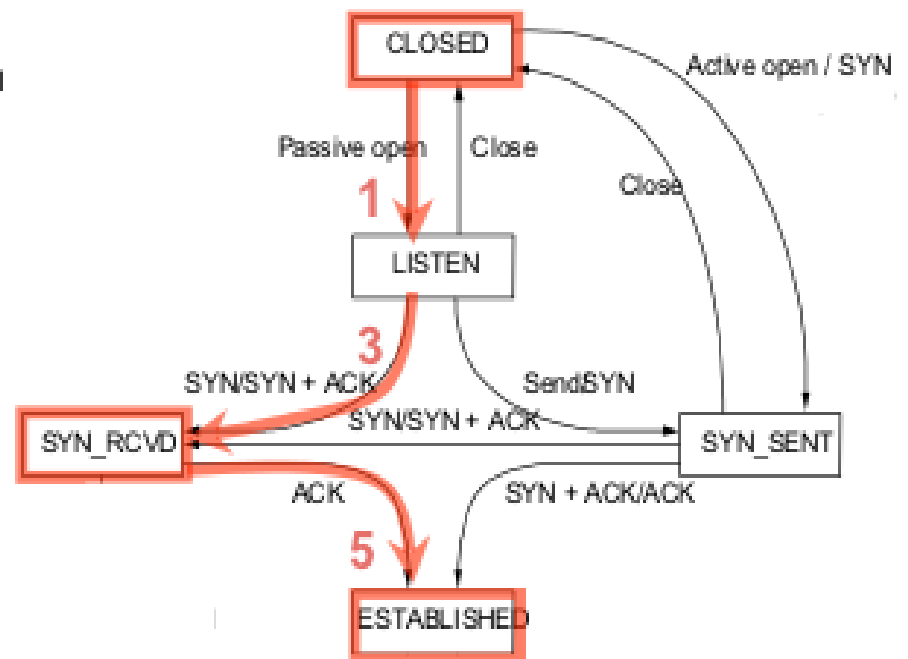


# Diagrama de estado TCP

## Cliente



## Servidor



# TCP – Finalizar una Conexión

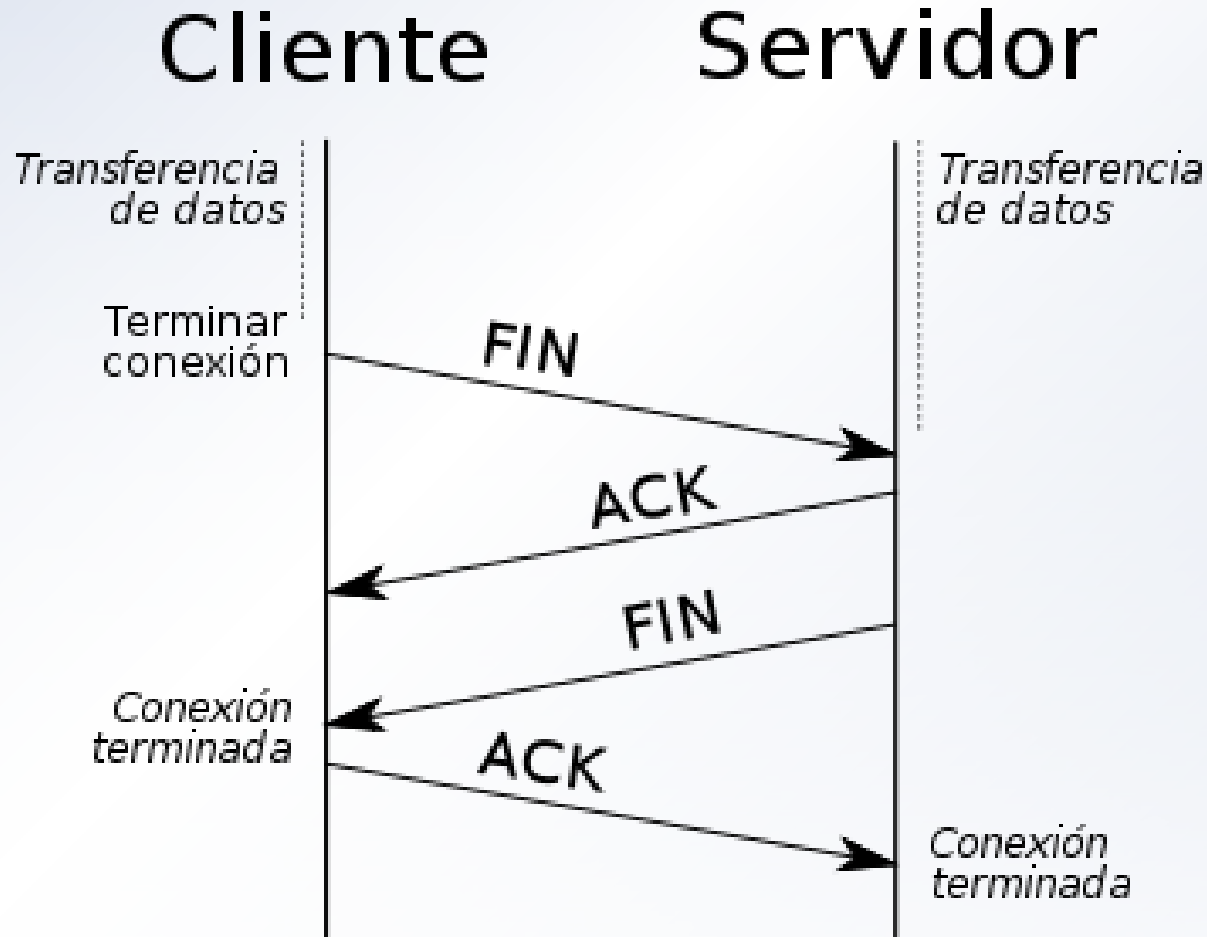
Fin de la conexión

Negociación de 4 pasos (4way handshake, FIN, ACK, FIN, ACK)

- Independiente de cada lado
- La conexión puede quedar “medio abierta”
- El lado que finalizó la conexión no podrá enviar más datos

En cualquier momento se puede enviar un RST

# Transporte Orientado a la Conexión



# ¿TCP o UDP?

Proveen diferentes servicios.

El programador, elije el protocolo que mejor satisface sus necesidades

Si se necesita un servicio de entrega confiable, TCP

Si se necesita un servicio de datagrama, UDP

Si se necesita eficiencia sobre circuitos de transporte extensos, TCP

Si se necesita eficiencia sobre redes rápidas con poca latencia, UDP

Si elige UDP y se necesita confiabilidad, entonces la aplicación debe proveer confiabilidad

# Puertos

Reservados 16 bits

Puertos bien conocidos

- 0 – 1023
- FTP (21) TCP
- DNS (53) UDP para transferencia de zona TCP
- SMTP (25) TCP
- HTTP (80) TCP
- IMAP (143) TCP
- NTP (123) UDP
- SNMP (161) UDP
- DHCP (67 y 68) UDP

Puertos Reservados

- 1024 – 65535

# Aplicaciones

netstat -patn

```
ngcena@nbgapa: ~  
netstat -pant  
(Not all processes could be identified, non-owned process info  
will not be shown, you would have to be root to see it all.)  
Active Internet connections (servers and established)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name  
tcp        0      0 0.0.0.0:62349          0.0.0.0:*               LISTEN      2676/wish  
tcp        0      0 0.0.0.0:111            0.0.0.0:*               LISTEN      -  
tcp        0      0 0.0.0.0:60628          0.0.0.0:*               LISTEN      -  
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      -  
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      -  
tcp        0      0 10.0.0.55:56873        190.220.135.226:143     ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:49508        190.114.198.101:993     ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:55951        65.54.48.136:1863      ESTABLISHED 2676/wish  
tcp        0      0 10.0.0.55:51985        74.125.134.16:993      ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:41883        200.43.77.4:993        ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:51974        74.125.134.16:993      ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:49968        190.114.198.101:993     ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:51978        74.125.134.16:993      ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:49506        190.114.198.101:993     ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:49509        190.114.198.101:993     ESTABLISHED 2931/thunderbird  
tcp        0      0 10.0.0.55:51934        74.125.134.16:993      ESTABLISHED 2931/thunderbird  
tcp6       0      0 :::1:631               :::*                   LISTEN      -  
tcp6       0      0 :::1:25                :::*                   LISTEN      -
```



# Aplicaciones

netstat -unpa

```
root@ ~ netstat -unpa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
udp        0      0 0.0.0.0:5353             0.0.0.0:*                793/avahi-daemon: r
udp        0      0 0.0.0.0:43882           0.0.0.0:*                793/avahi-daemon: r
udp        0      0 10.0.0.10:53            0.0.0.0:*                1073/named
udp        0      0 127.0.0.1:53            0.0.0.0:*                1073/named
udp        0      0 0.0.0.0:68              0.0.0.0:*                5529/dhclient
udp6       0      0 :::5353                 :::*                    793/avahi-daemon: r
udp6       0      0 :::34106                :::*                    793/avahi-daemon: r
udp6       0      0 ::1:36798               ::1:36798              ESTABLISHED 1975/postmaster
udp6       0      0 :::53                   :::*                    1073/named
```

# Palabras Claves

Multiplexación Demultiplexación

Socket

UDP

TCP

3-way handshake

Control de Flujo

Slow-Star

4-way handshake

Puertos

netstat