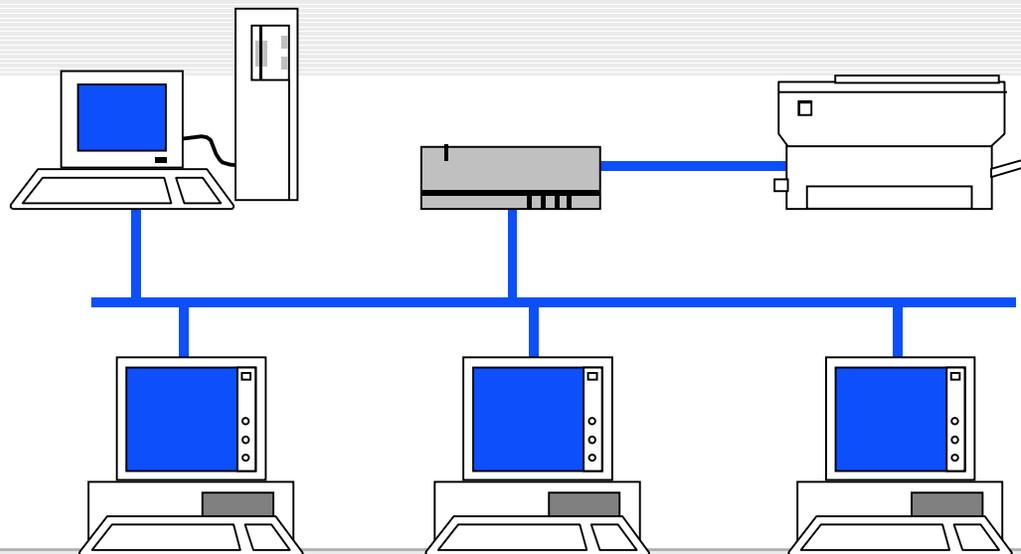


Redes de Computadores

Capa de Transporte TCP





Características

Objetivo

Proveer un “transporte” de datos confiable y eficiente entre una máquina origen y una máquina destino, independiente de las redes físicas o lógicas que deban cruzar.

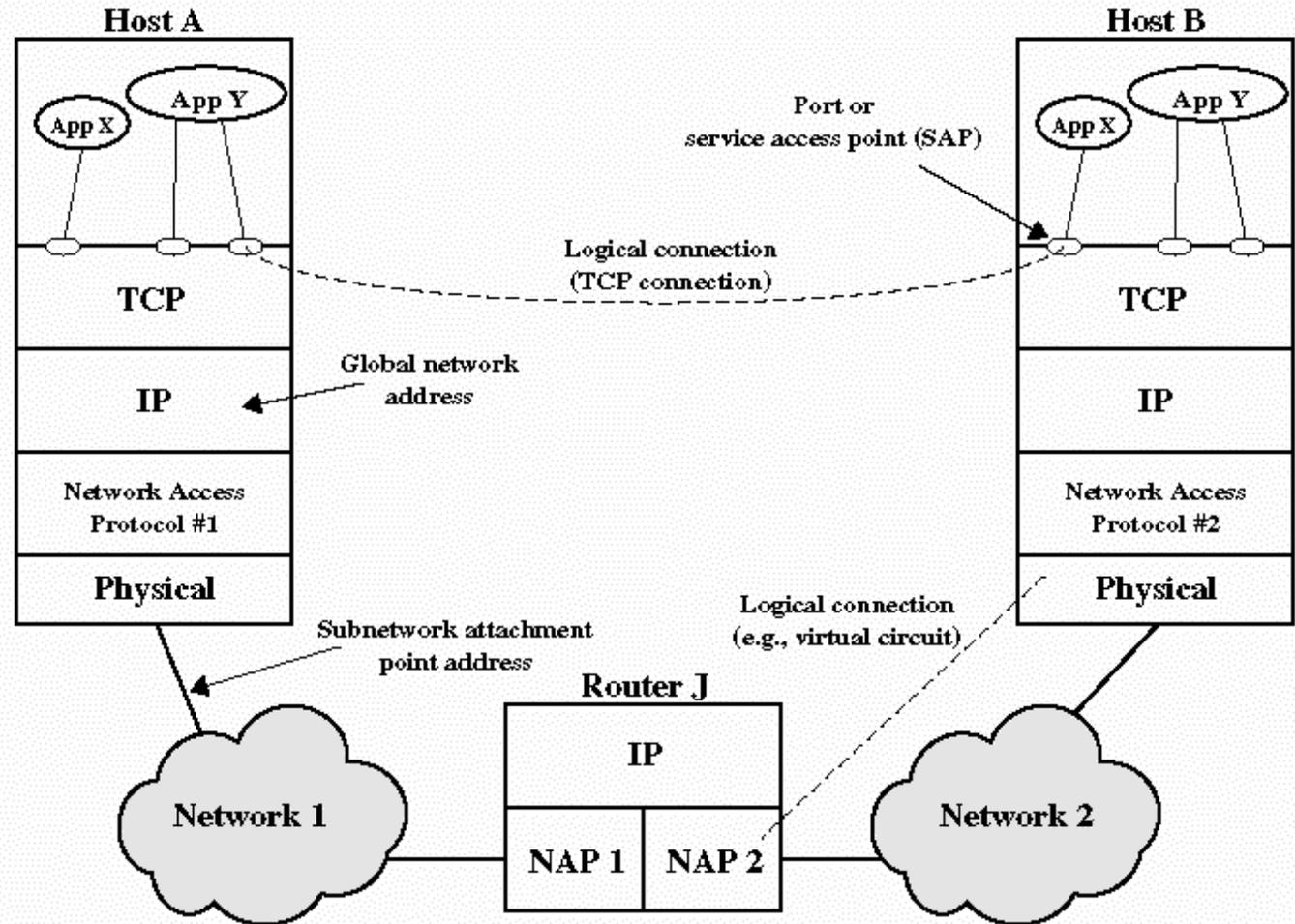
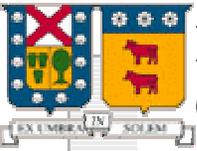


Figure 12.8 TCP/IP Concepts



Medidas de QOS

Connection establishment delay

Connection establishment failure probability

Throughput

Transit delay

■ CED: Tiempo que transcurre entre que se solicita una conexión y se recibe la confirmación de conexión lista

■ CEFP: Es la probabilidad de que no sea posible establecer una conexión dentro del máximo CED establecido, sea por congestión u otro motivo

■ Throughput: # bytes de data de usuario transferido por segundo, medido en un cierto intervalo. Es medido en ambas direcciones

■ TD: tiempo entre la emisión de un mensaje (TPDU) y la recepción de éste.



Medidas de QOS

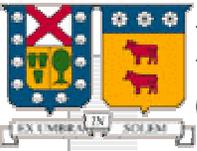
Residual error ratio
Protection
Priority
Resilience

- RER: tasa de mensajes malos v/s mensajes totales enviados
- Protección: parámetro que indica si la conexión es o no “secreto” y por ende solicita nivel de encriptación
- Prioridad: indica la importancia de la conexión
- Resiliencia: indica la probabilidad de que la capa de transporte termine la conexión debido a problemas internos o congestión.



Medidas de QOS

- Los parámetros de QOS (Quality of Service, Calidad de Servicio) son especificados por la máquina que solicita la conexión
- Si el receptor de la conexión puede cumplir con dichos parámetros, entonces se establece la conexión y se mantienen durante toda la conexión
- Si el receptor no puede cumplir los parámetros, rechaza la conexión
- El emisor puede especificar 2 valores para el mismo parámetro: un mínimo y un deseable
- Este proceso se llama “autonegociación de opciones”.



TDPU

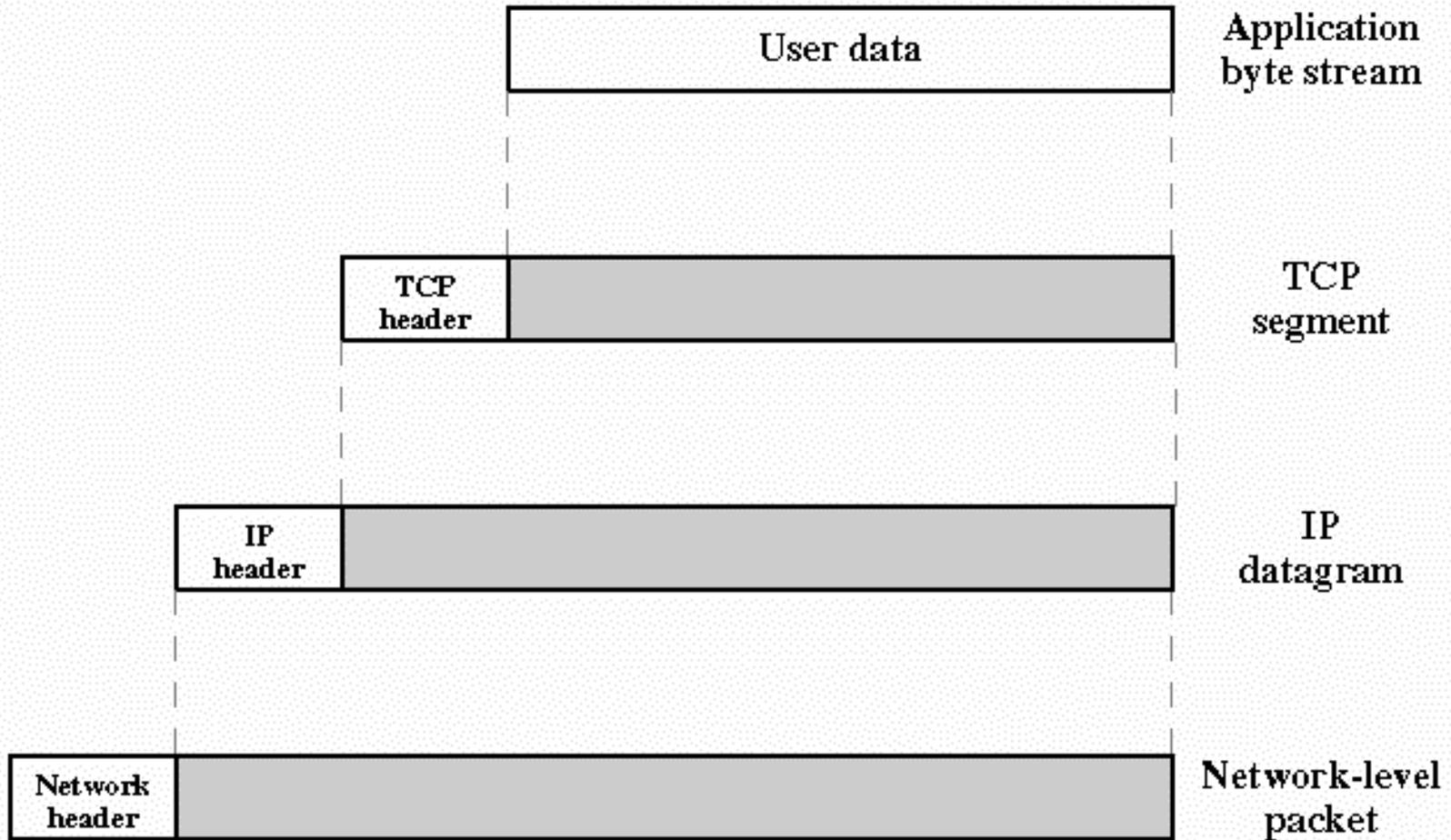


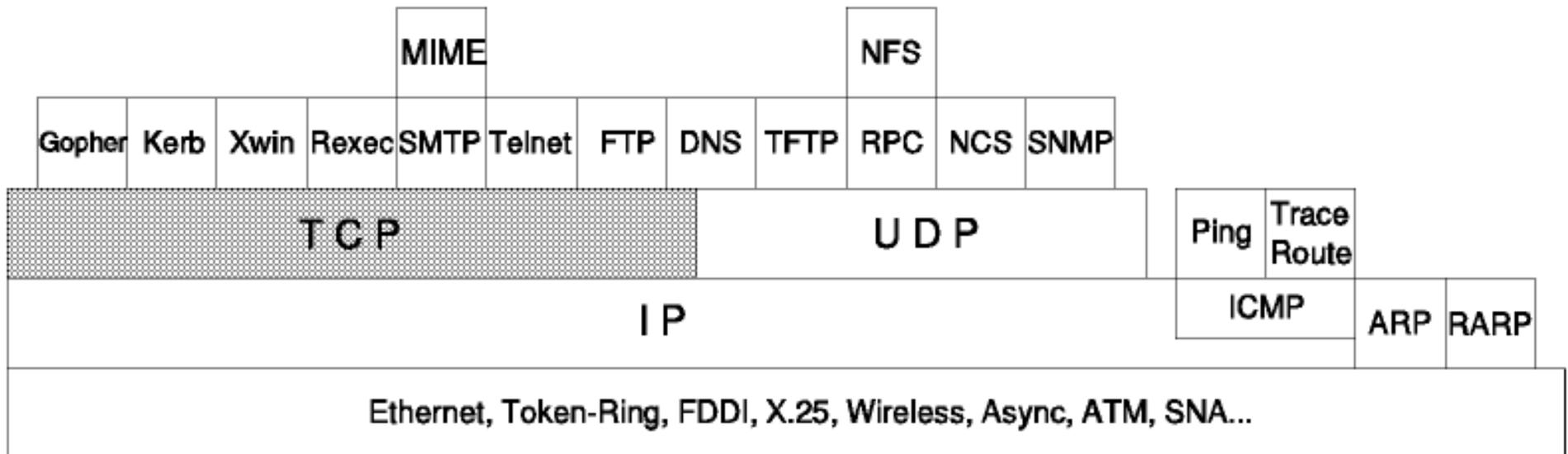
Figure 12.9 Protocol Data Units in the TCP/IP Architecture



Capa de Transporte en Internet

2 protocolos:

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)





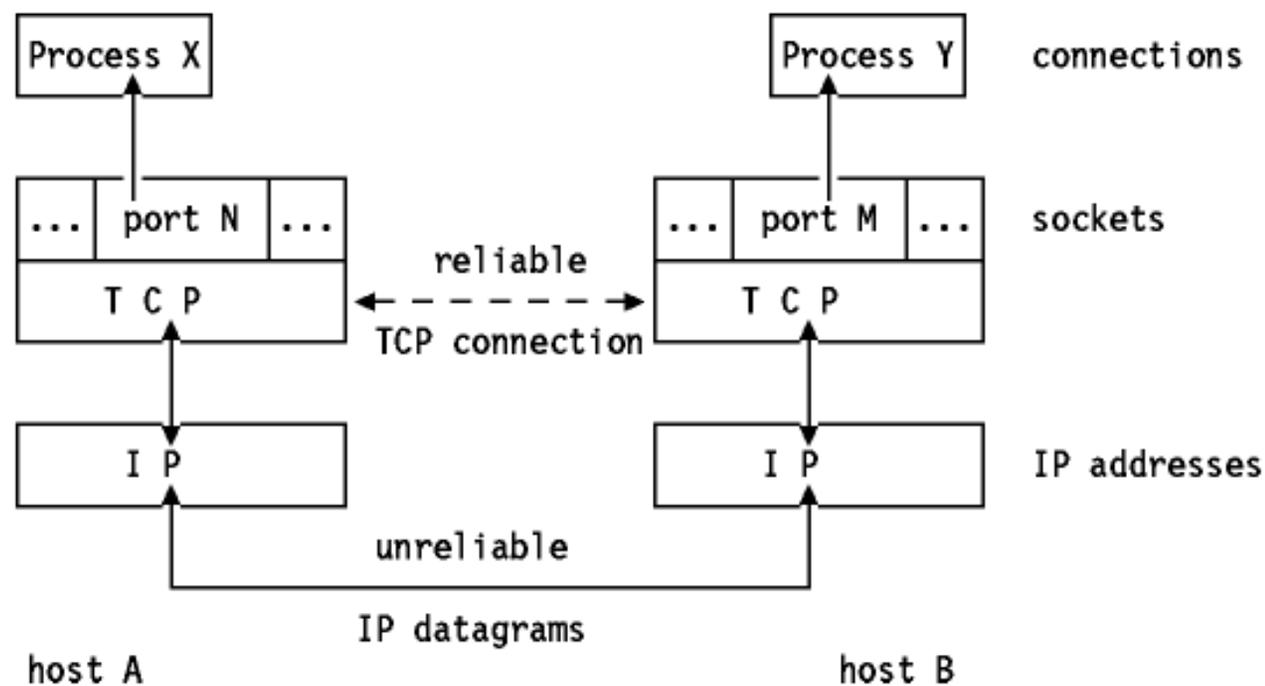
TCP (RFC 793, 1122, 1323)

- servicio orientado a la conexión
- entrega confiable de datos
- transmite datos en full duplex
- asigna un #secuencia a cada byte, por lo que espera un ACK del receptor indicando el último #secuencia correctamente recibido
- posee control de flujo independiente para ambos sentidos de TX (el RX indica en los mismos ACK, hasta cuál #secuencia puede recibir)
- posee un mecanismo de control de congestión
- IP no garantiza entrega de paquetes; “tarea” que asume TCP
- TX datos a la conexión en forma de segmentos o mensajes TDPU y el RX lo lee byte a byte (no existe la visión de TDPU), igual como se operara con un archivo
- El tamaño del mensaje está restringido por IP (65536 bytes de data) y por la MTU de la Capa 2, para no sufrir segmentación en la red local
- permite multiplexión con un mismo RX (queda identificado como otra conexión, otro par de sockets).



TCP ports & sockets

- 2 procesos se comunican usando TCP Sockets
- Cada extremo de una conexión TCP posee un socket para esa conexión
- socket = <dirección_IP , port>
- una conexión queda totalmente definida y es única, especificando el par de sockets: (IP_local, port_local, IP_remoto, port_remoto).





TCP ports & sockets

- Los ports menores a 1024 se ocupan principalmente para proveer servicios de red
- Los ports desde 1024 a 65536 son ocupados en base a demanda y luego son liberados
- Port conocidos o “well known” (usualmente menores a 256) (RFC 1700)
 - 20 y 21 FTP command y FTP transfer
 - 22 ssh
 - 23 telnet
 - 25 SMTP (email)
 - 53 DNS
 - 80 WWW
 - 110 POP3
 - 119 NNTP (news)
 - 6000 X (despliegue gráfico)



Conexiones en un PC

netstat -rn

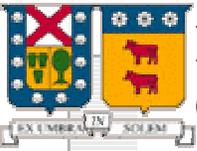
```

MS-DOS
7 x 11
Rutas activas:
Dirección de red      Máscara de red  Puerta de enlace      Interfaz  Métrica
0.0.0.0              0.0.0.0        200.1.21.1            200.1.21.50  1
127.0.0.0           255.0.0.0      127.0.0.1             127.0.0.1    1
200.1.21.0          255.255.255.192 200.1.21.50           200.1.21.50  1
200.1.21.50         255.255.255.255 127.0.0.1             127.0.0.1    1
200.1.21.255       255.255.255.255 200.1.21.50           200.1.21.50  1
224.0.0.0           224.0.0.0      200.1.21.50           200.1.21.50  1
255.255.255.255    255.255.255.255 200.1.21.50           0.0.0.0      1

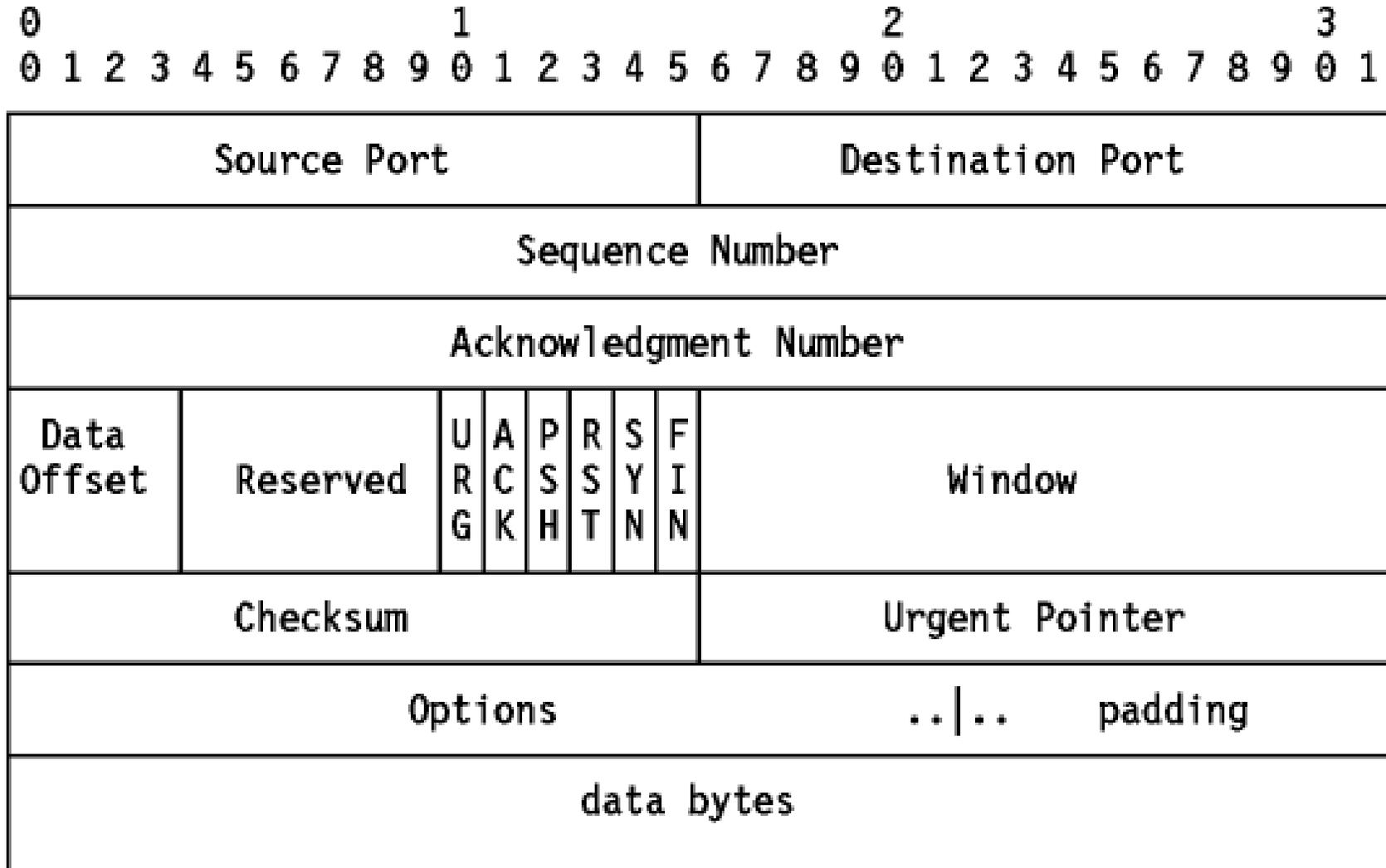
Conexiones activas
Proto  Dirección local      Dirección remota      Estado
TCP    127.0.0.1:1655       127.0.0.1:80          TIME_WAIT
TCP    200.1.21.50:1044     200.1.21.43:139      ESTABLISHED
TCP    200.1.21.50:1103     146.83.198.8:80      CLOSE_WAIT
TCP    200.1.21.50:1105     146.83.198.8:80      CLOSE_WAIT
TCP    200.1.21.50:1106     146.83.198.8:80      CLOSE_WAIT
TCP    200.1.21.50:1107     146.83.198.8:80      CLOSE_WAIT
TCP    200.1.21.50:1108     146.83.198.8:80      CLOSE_WAIT
TCP    200.1.21.50:1111     146.83.198.8:80      CLOSE_WAIT

C:\WINDOWS>

```



Header TCP





Header TCP

- Source & Destination Port: puertos TCP local y remoto
- Sequence number: identifica el a cuál byte del flujo total corresponde el primer byte de este mensaje
- Ack number: contiene el próximo #secuencia que el emisor del ack espera recibir. (#secuencia del último byte recibido correctamente +1)
- Flag ACK: indica que el campo de ACK number es válido y que el RX debe “mirarlo”
- Data Offset o Header Length: indica el largo del encabezado en palabras de 32 bits
- Flag URG: indica que en este mensaje existe data urgente y que el “Urgent Pointer” está en uso
- Urgent Pointer: Indica el offset desde donde se encuentra la data urgente.



Header TCP

- Flags SYN y FIN: usando para establecer y cerrar una conexión
- Flag URG: indica que en este mensaje existe data urgente
- Flag PUSH: indica que la data debe ser entregada a la aplicación inmediatamente y no puesta en un buffer
- Flag RESET: el receptor se ha “confundido” y por ende desea abortar la conexión o simplemente rechaza el pedido de conexión

- Checksum: cubre el header TCP y la data

- Option:
 - permite negociar el MSS (Maximum Segment Size). Ambas partes indican el tamaño máximo de segmento TCP que desean recibir
 - permite seleccionar selective repeat en vez de go-back-n (RFC 1106)



MSS Maximum Segment Size

- MSS por defecto 536 octetos correspondiente al tamaño por defecto del datagrama IP, 576, menos el tamaño standard de los encabezados IP y TCP
- El MSS se elige tal que genere el datagrama IP igual al mínimo MTU existente en el camino entre fuente y destino
- Un MSS muy pequeño genera overhead. Transmitir sólo un octeto de data utiliza, a lo más, el 1/41 del BW de la capa contigua
- Un MSS muy grande genera datagramas grandes que, al pasar por redes con MTU pequeño, son fragmentados.



MSS Maximum Segment Size

- No se puede retransmitir un fragmento de datagrama ni enviar un ack hasta tener el datagrama completo, por lo que disminuye la probabilidad de que un datagrama llegue correcto....

Disminuye el throughput

- El tamaño óptimo de segmento es aquél que genere un datagrama lo más grande posible sin necesidad de fragmentación a lo largo de su camino entre la fuente y el destino

- Problema:

Los routers que ocupan asignación dinámica de rutas hacen que el camino que siguen los datagramas varíe dinámicamente, ocasionando fragmentación. Además el tamaño óptimo también depende de los headers de los protocolos de más bajo nivel.



Establecimiento de la Conexión

- Caso (a): Host 1 inicia conexión a Host 2
- Caso (b): Ambos Host inician conexión en justo los mismos #secuencia. Sólo se establece 2 conexión (x,y)

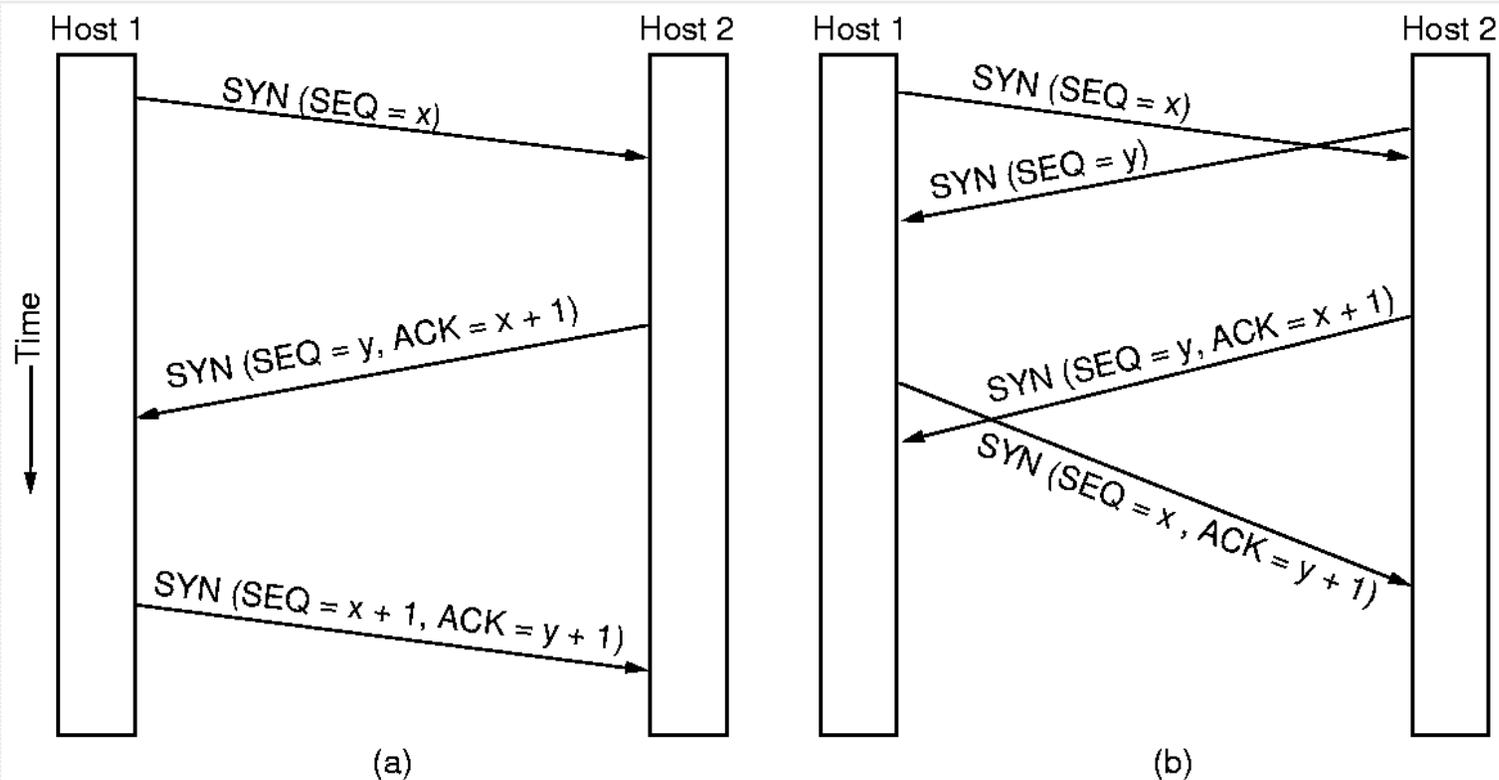


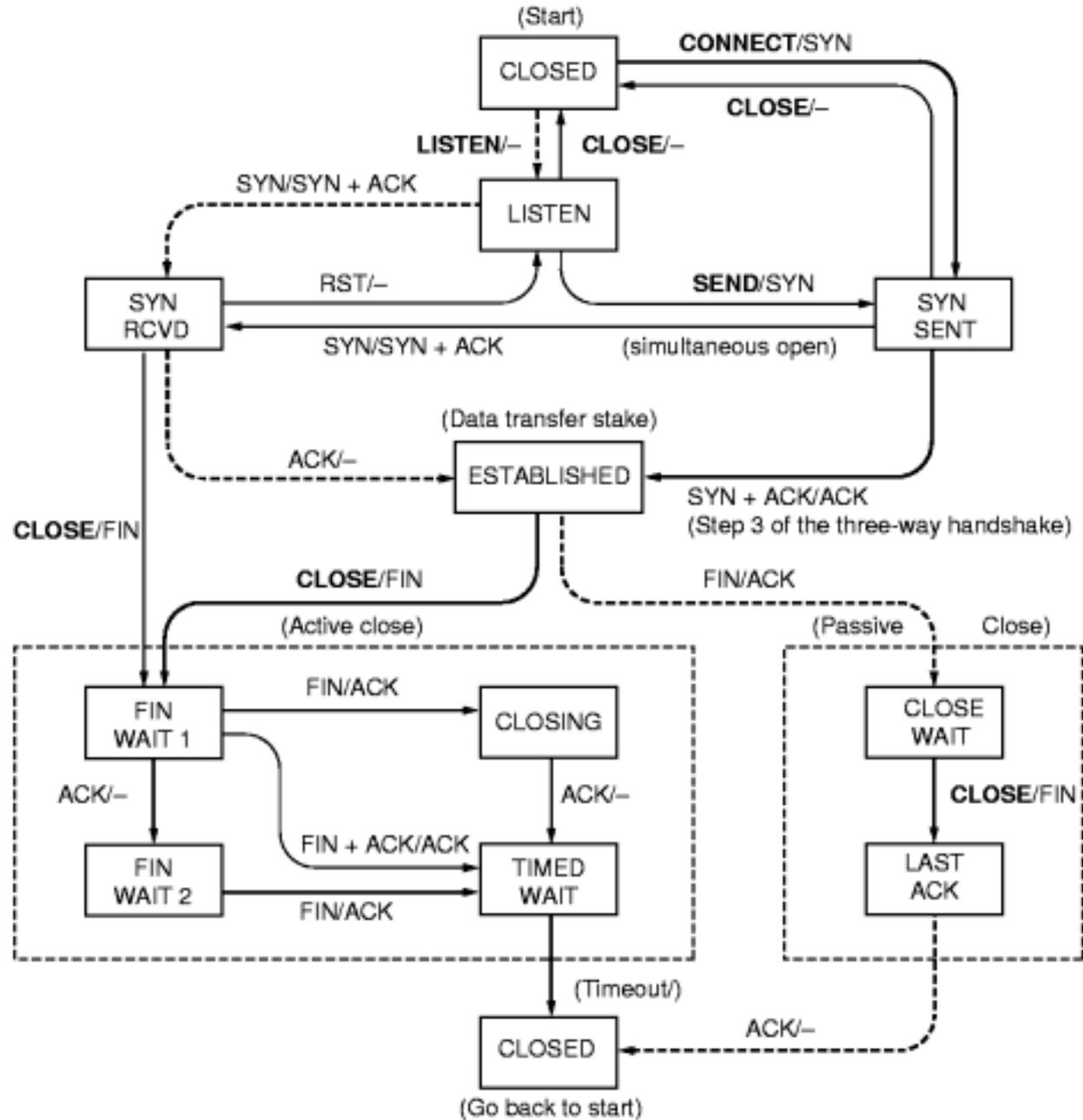


Diagrama de Estados TCP

evento/acción

cliente: línea gruesa

servidor: línea gruesa
segmentada





Control de Flujo

Los ACKs no son directamente ligados al corrimiento de la ventana deslizante

RX indica al TX cuanto puede recibir

