TCP sockets

Client must contact server

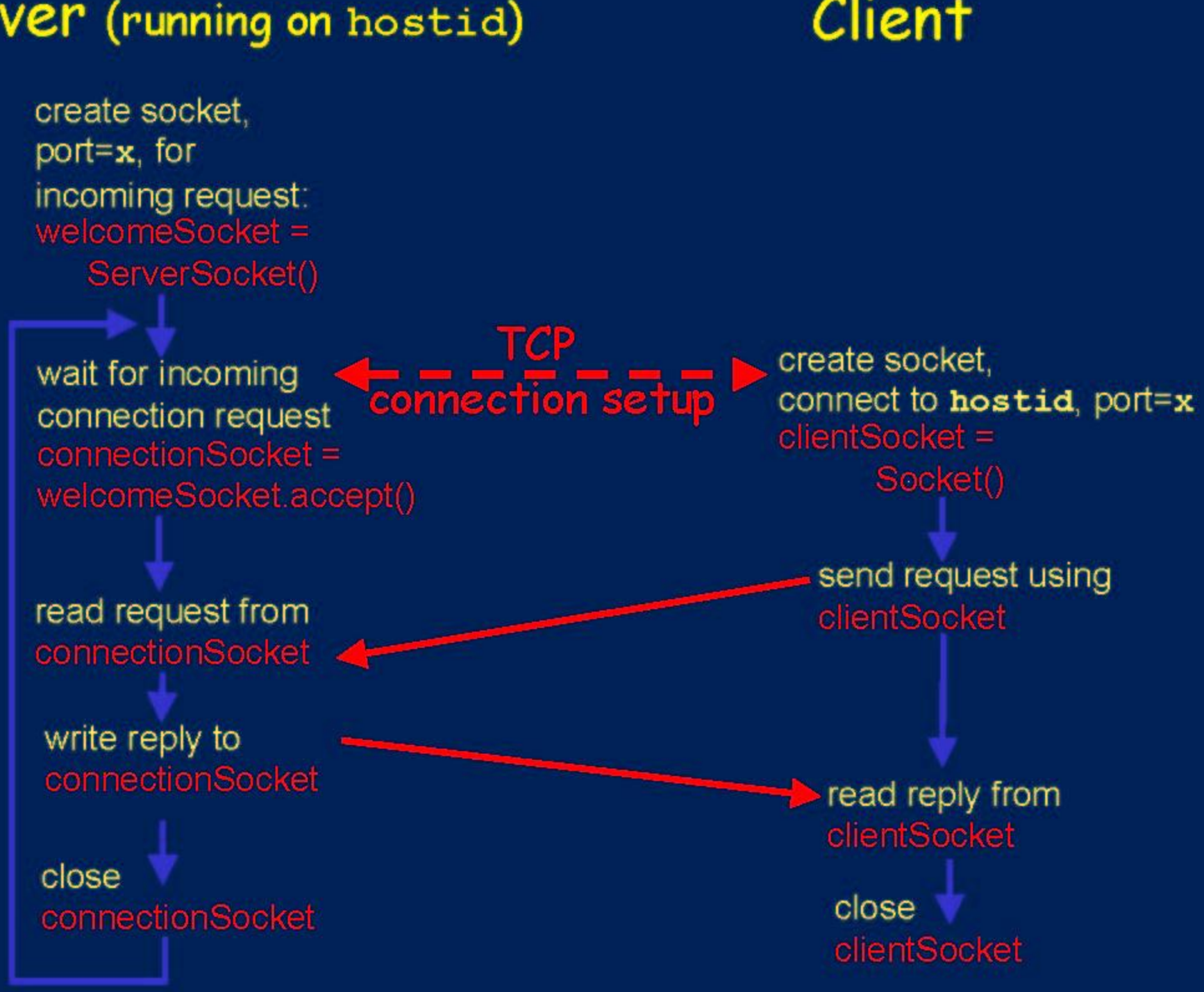
- server process must first be running
- server must have created socket (door) that welcomes client's contact

Client contacts server by:

- creating client-local TCP socket
- specifying IP address, port number of server process

- When client creates socket: client TCP establishes connection to server TCP
- When contacted by client, server TCP creates new socket for server process to communicate with client
  - allows server to talk with multiple clients

Client/server socket interaction: TCP



UDP Sockets

UDP: no "connection" between client and server

- **no handshaking**
- **sender explicitly attaches IP address and port of destination**
- **server must extract IP address, port of sender from received datagram**

UDP: transmitted data may be received out of order, or lost

application viewpoint

UDP provides unreliable transfer of groups of bytes ("datagrams") between client and server

Client/server socket interaction: UDP

Server (running on `hostid`)

create socket,
port=**x**, for
incoming request:
serverSocket =
DatagramSocket()

read request from
serverSocket

write reply to
serverSocket
specifying client
host address,
port umber

Client

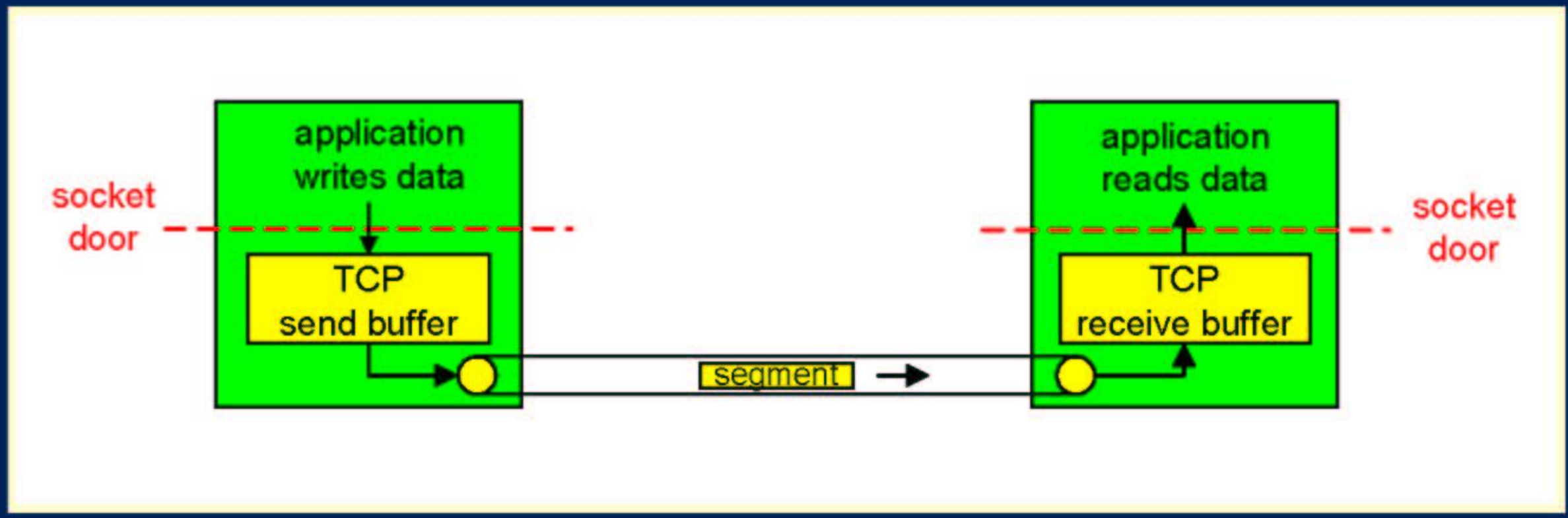
create socket,
clientSocket =
DatagramSocket()

Create, address (**`hostid, port=x,`**
send datagram request
using clientSocket

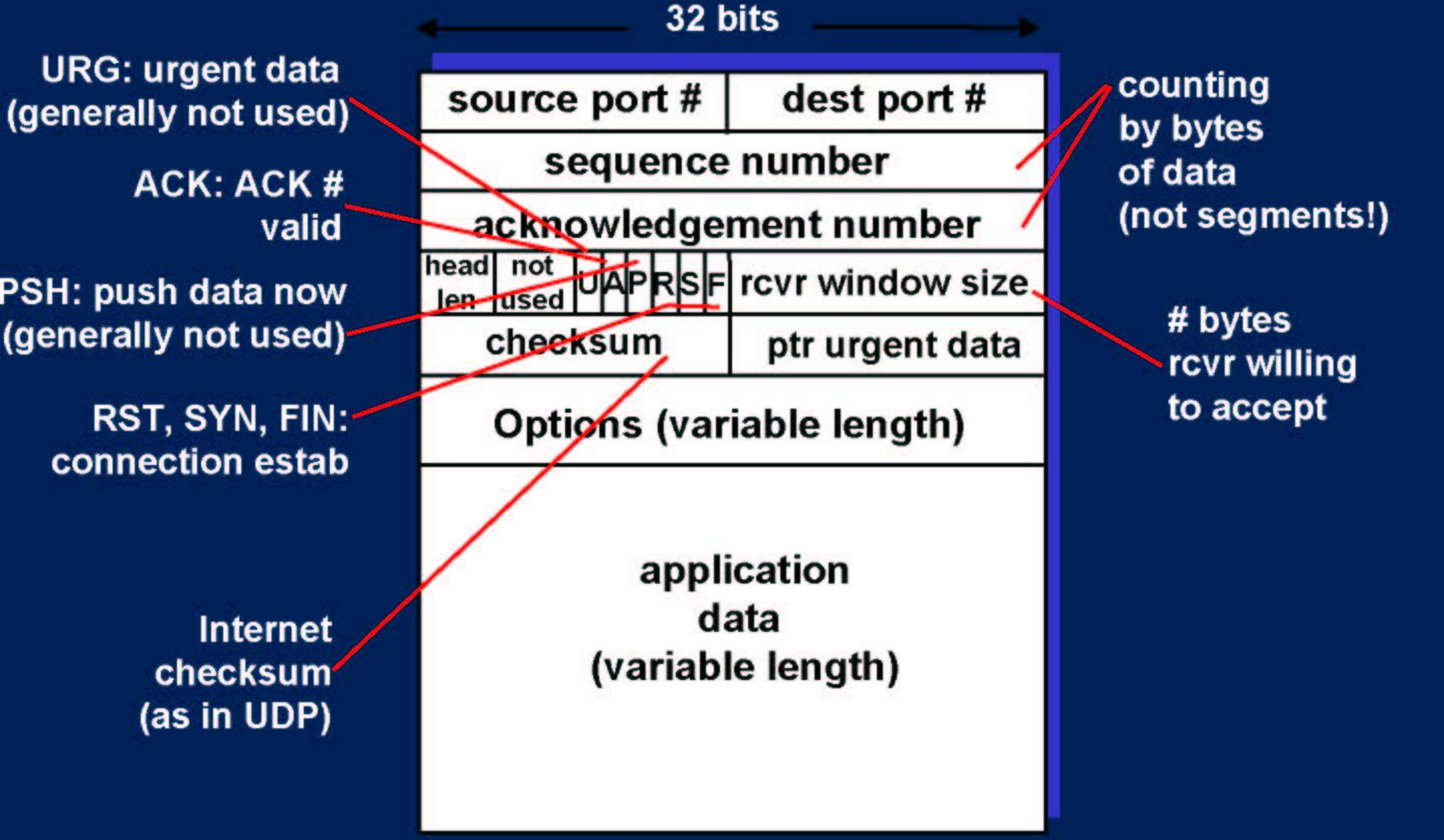
read reply from
clientSocket

close
clientSocket

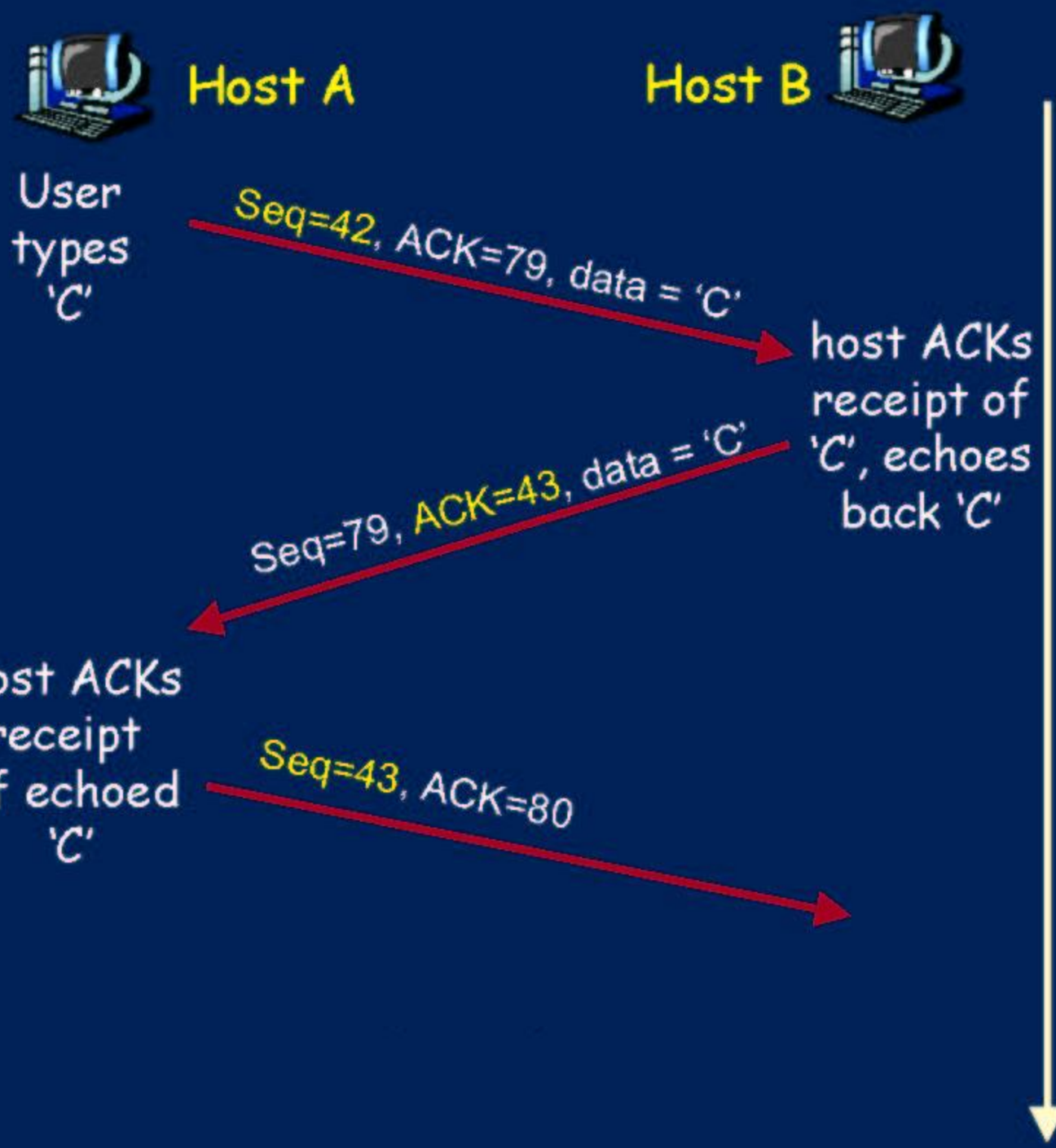
TCP: Overview



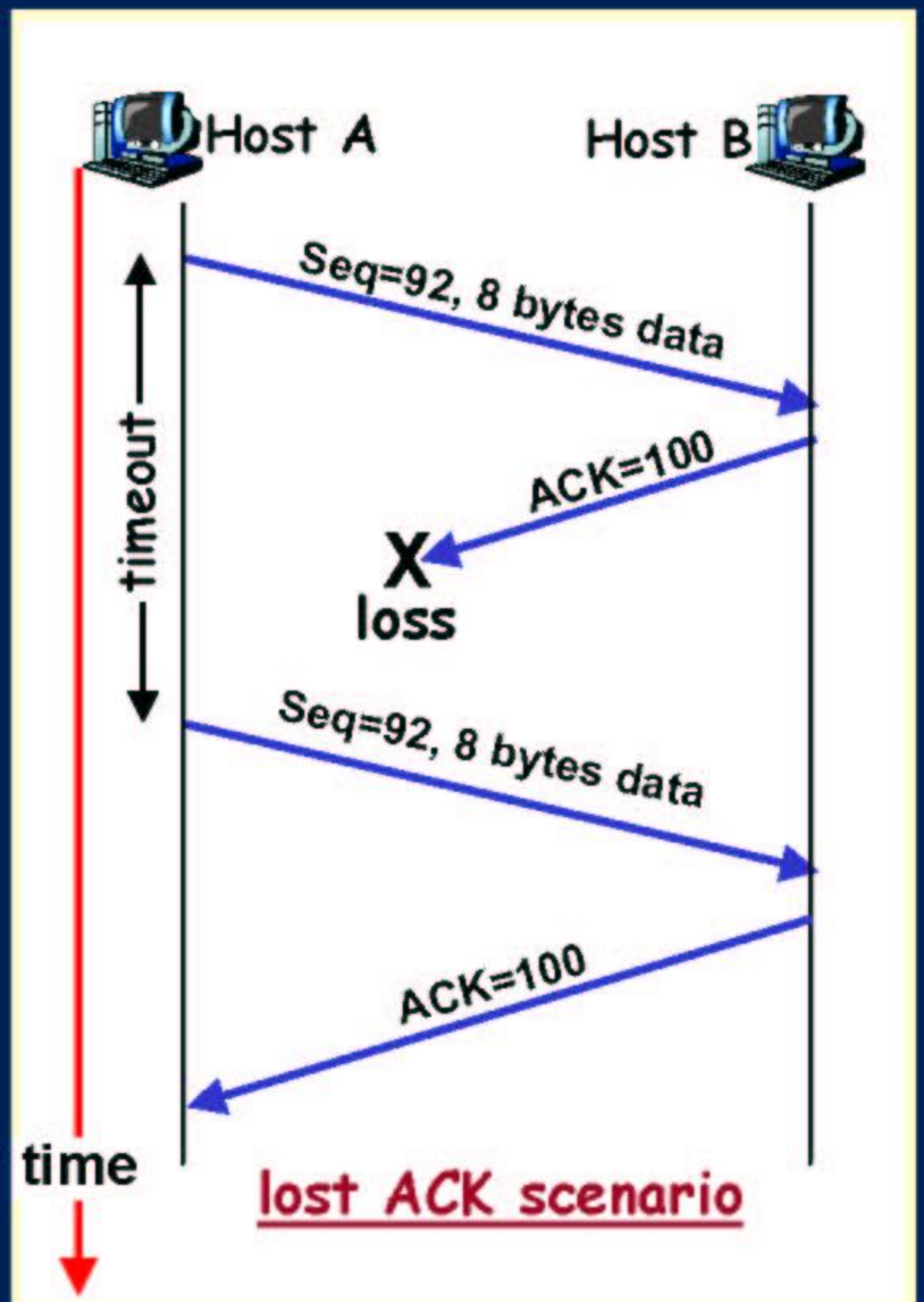
TCP segment structure



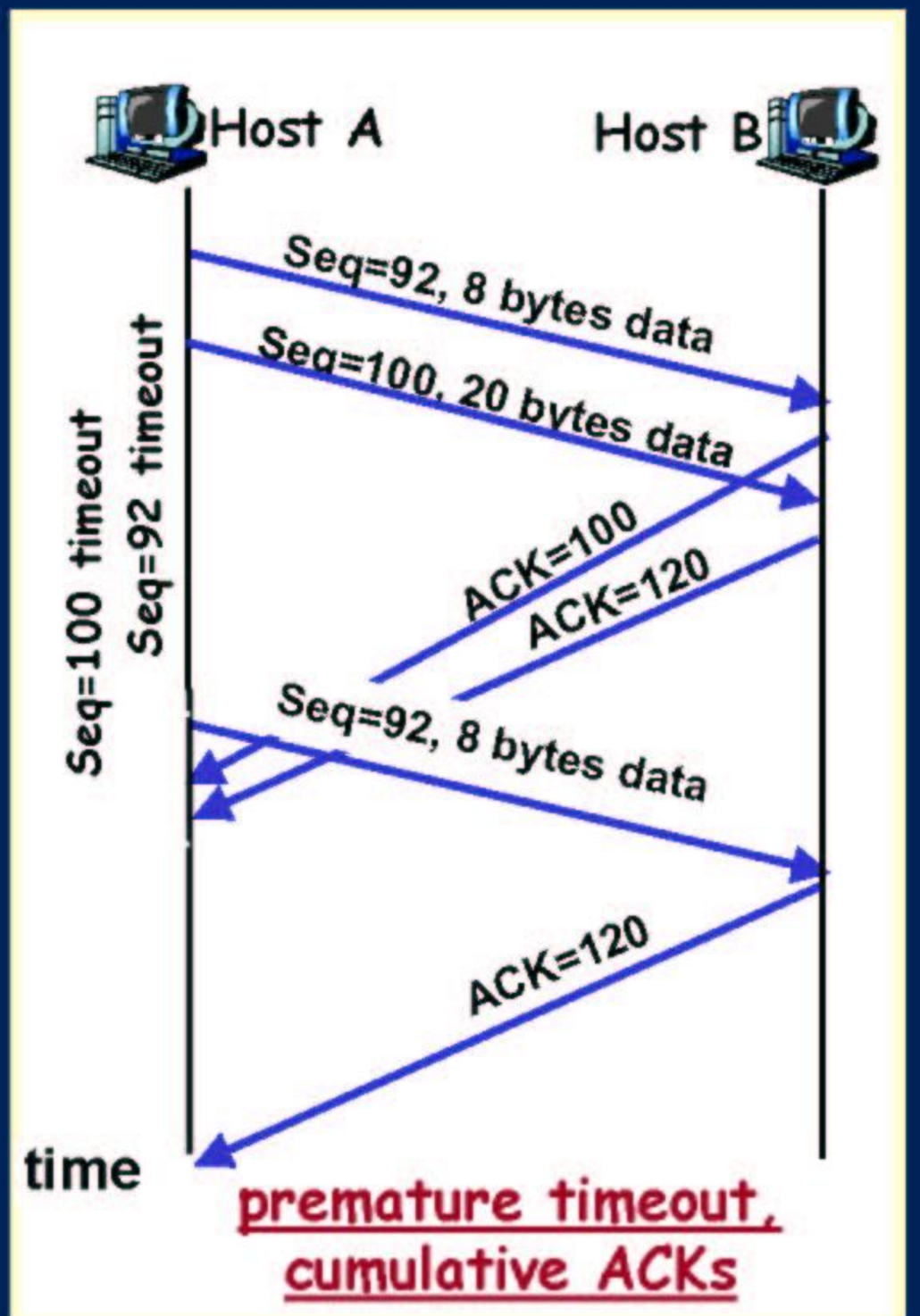
TCP Seq. #'s and ACKs (II)



TCP: retransmission scenarios



lost ACK scenario



premature timeout, cumulative ACKs

TCP Connection Management - 4