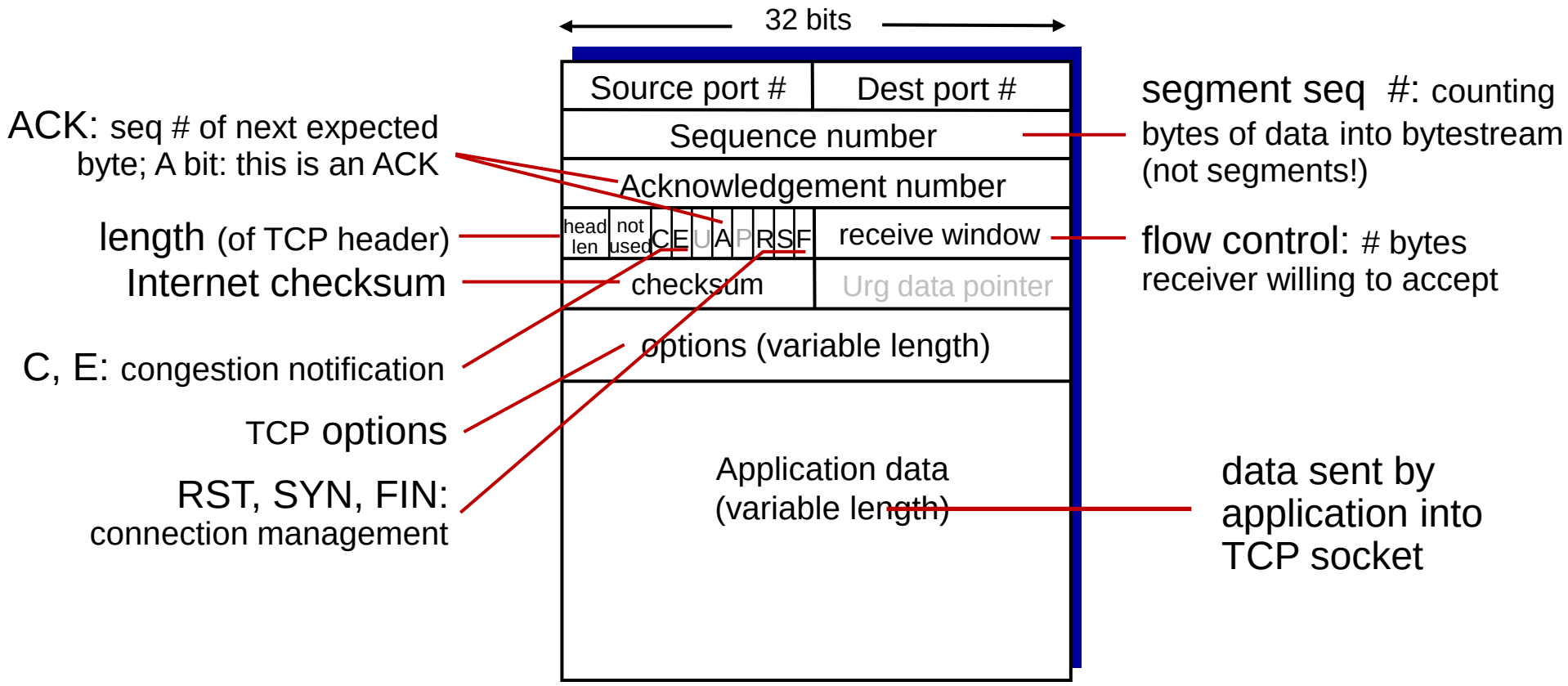
Computer Networks

TCP Connection Management

Amitangshu Pal
Computer Science and Engineering
IIT Kanpur

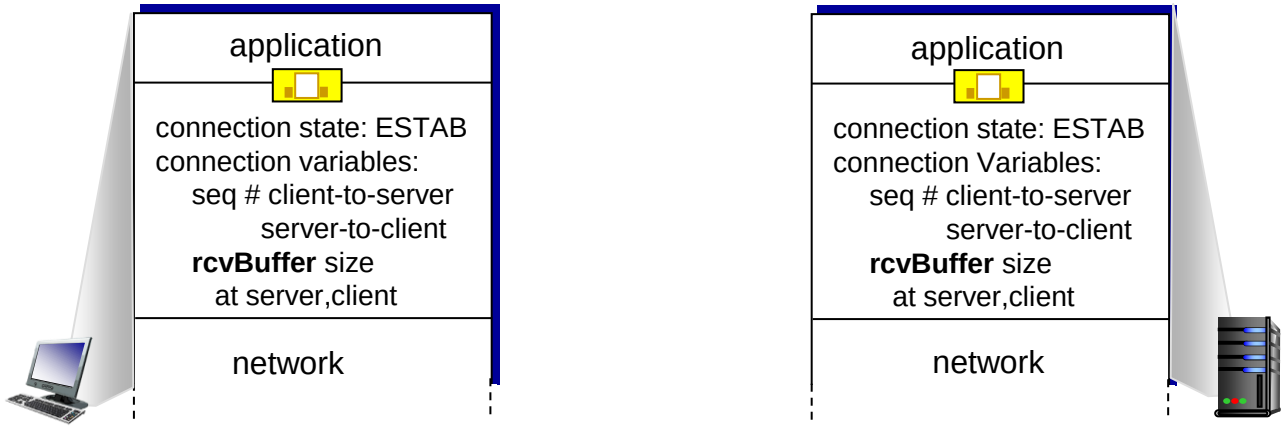
TCP Segment Structure



TCP Connection Management

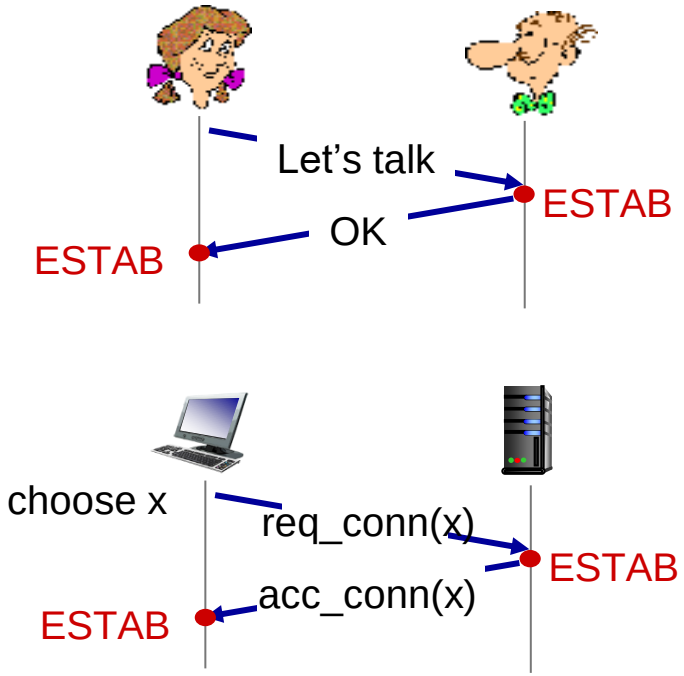
Before exchanging data, sender/receiver "handshake":

- Agree to establish connection (each knowing the other willing to establish connection)
- Agree on connection parameters (e.g., starting seq #s)



Agreeing to Establish a Connection

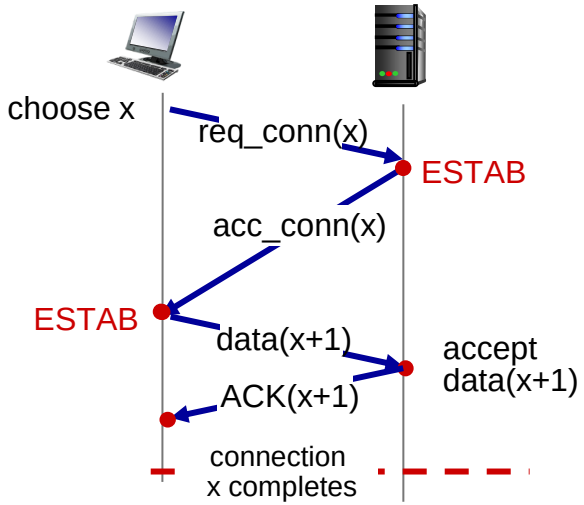
2-way handshake:



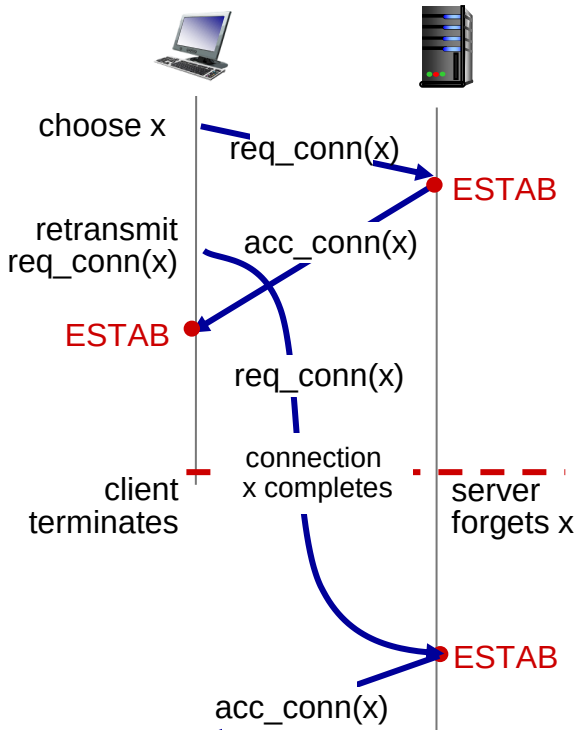
Will 2-way handshake always work in network?

- Variable delays
- Retransmitted messages (e.g. req_conn(x)) due to message loss
- Message reordering

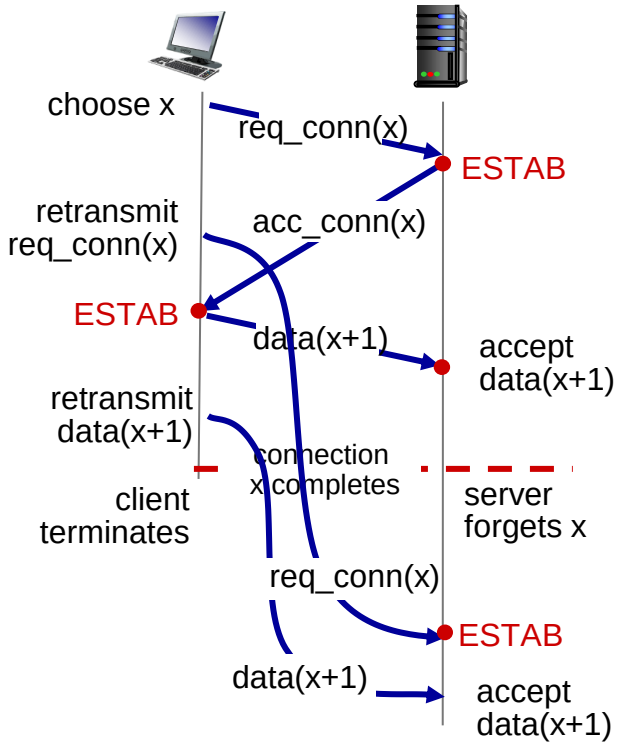
2-way Handshake Scenarios



No problem!

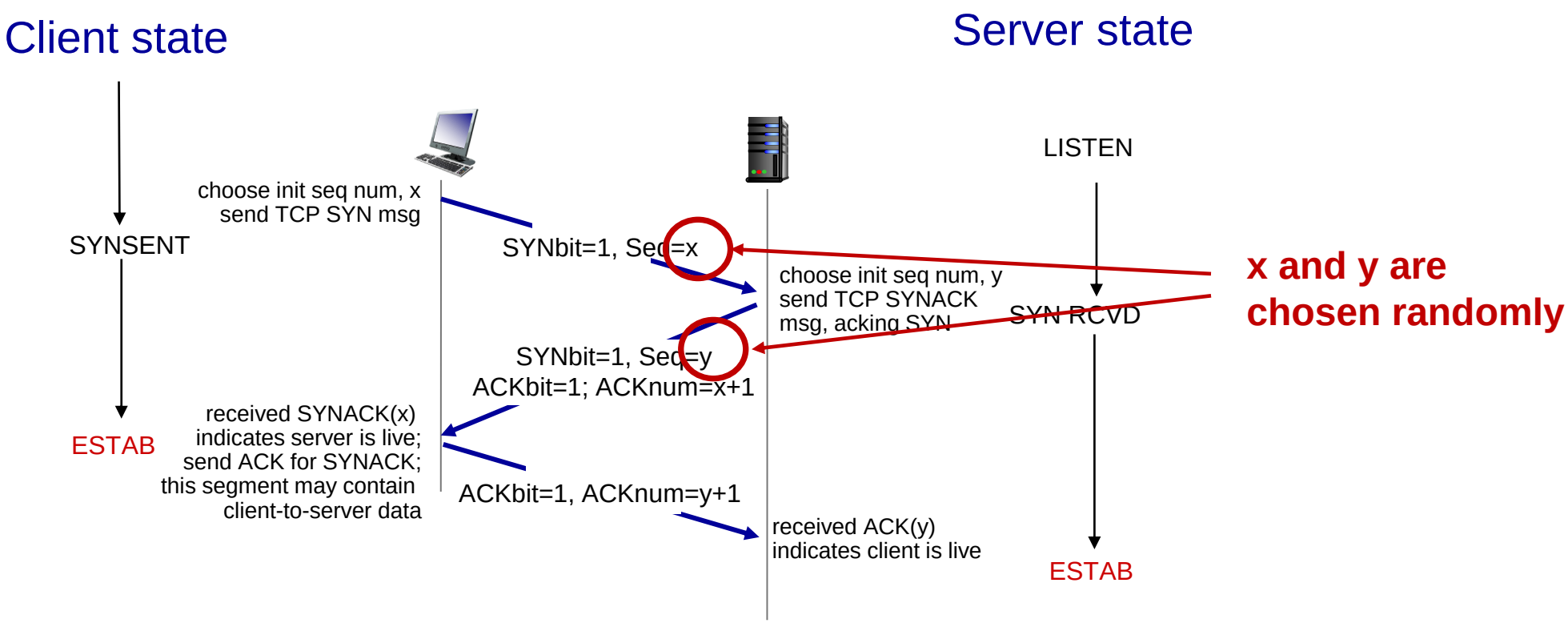


Problem: half open connection! (no client)

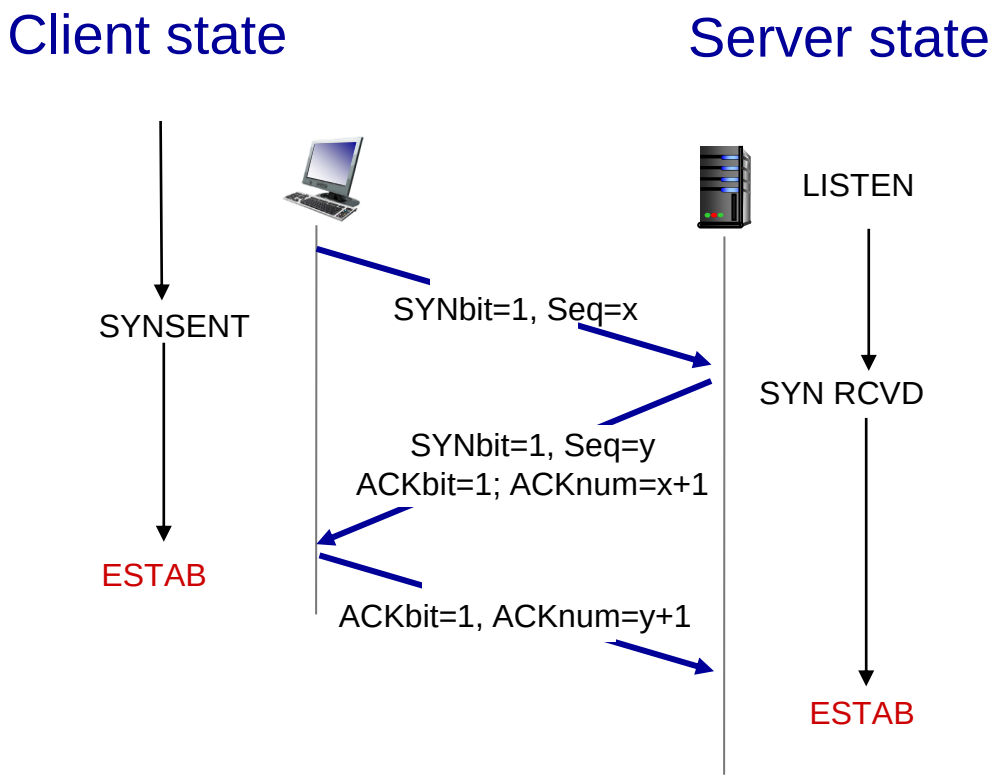


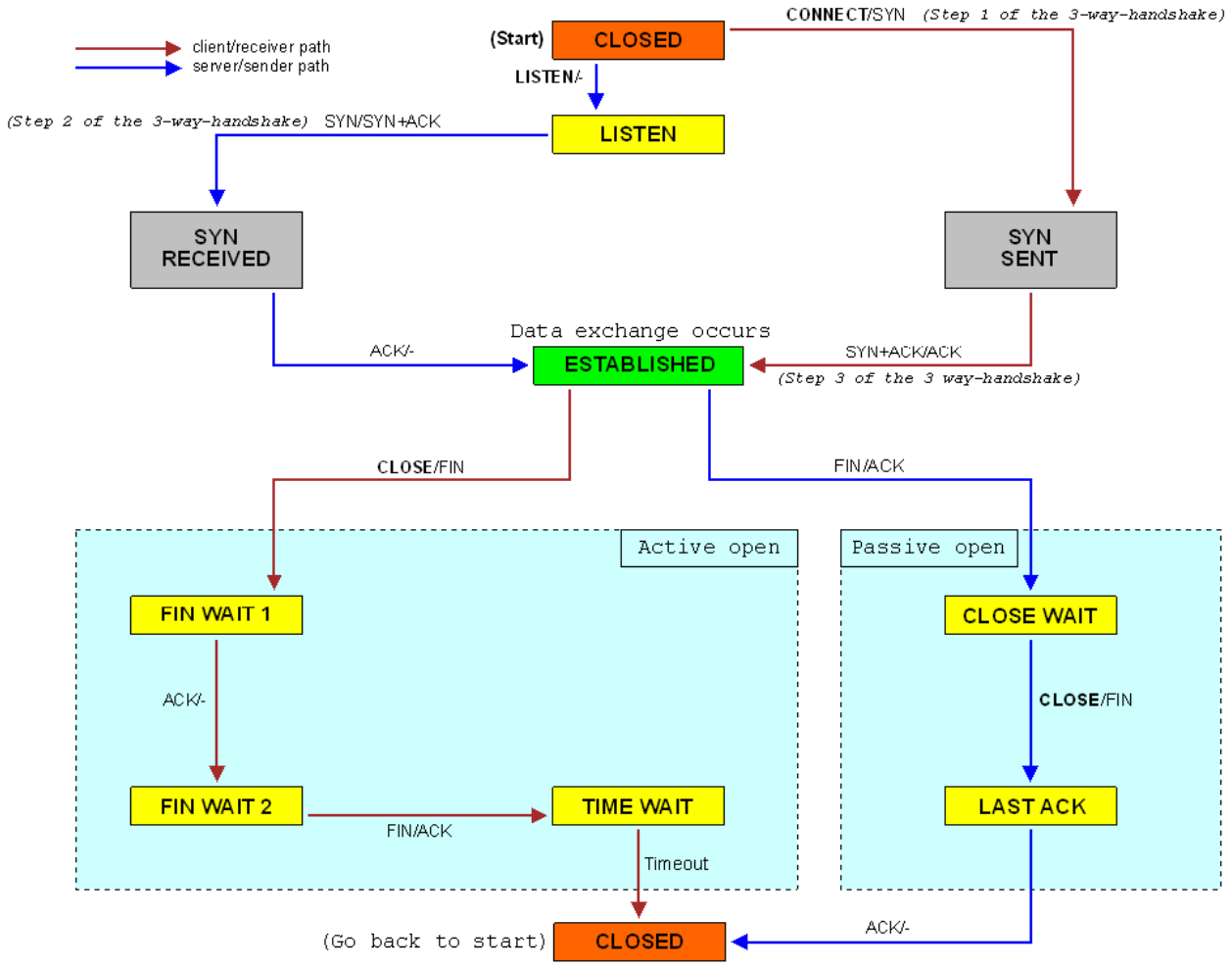
Problem: dup data accepted!

TCP 3-way Handshake



TCP State Diagram





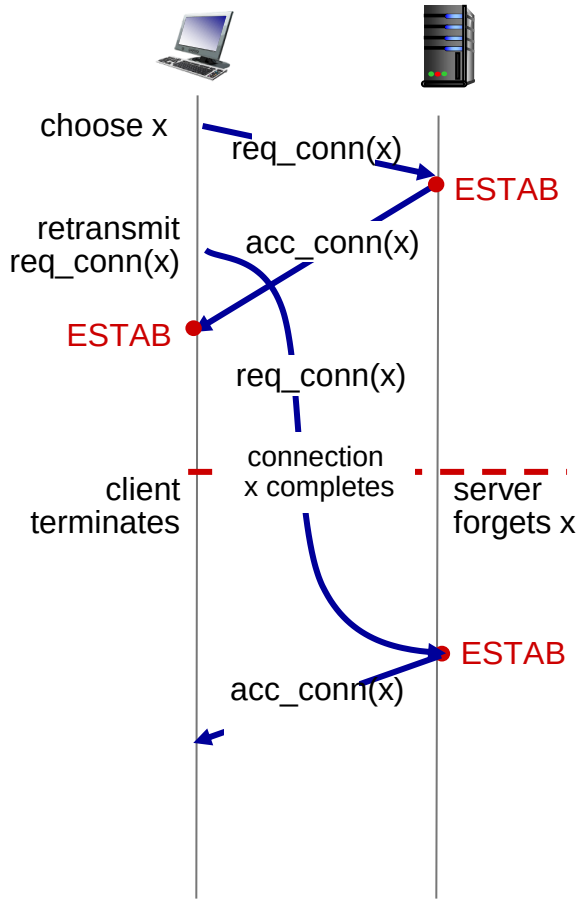
Src: https://en.wikipedia.org/wiki/File:Tcp_state_diagram.png

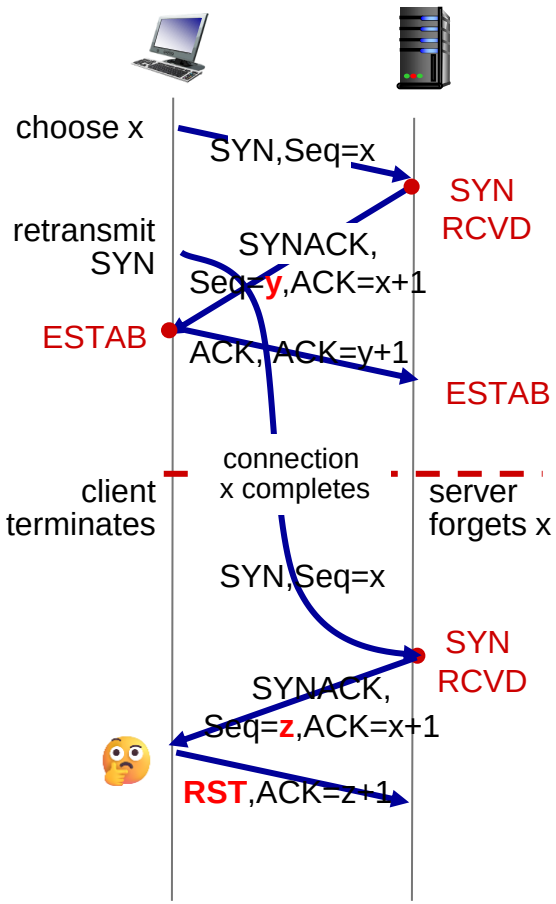
TCP 3-way Handshake

RST: Abort connection

- Receiver is confused
  - Unexpected sequence number
  - Port number or IP address do not match with any ongoing sockets

How it will solve the “half-open connection” problem?



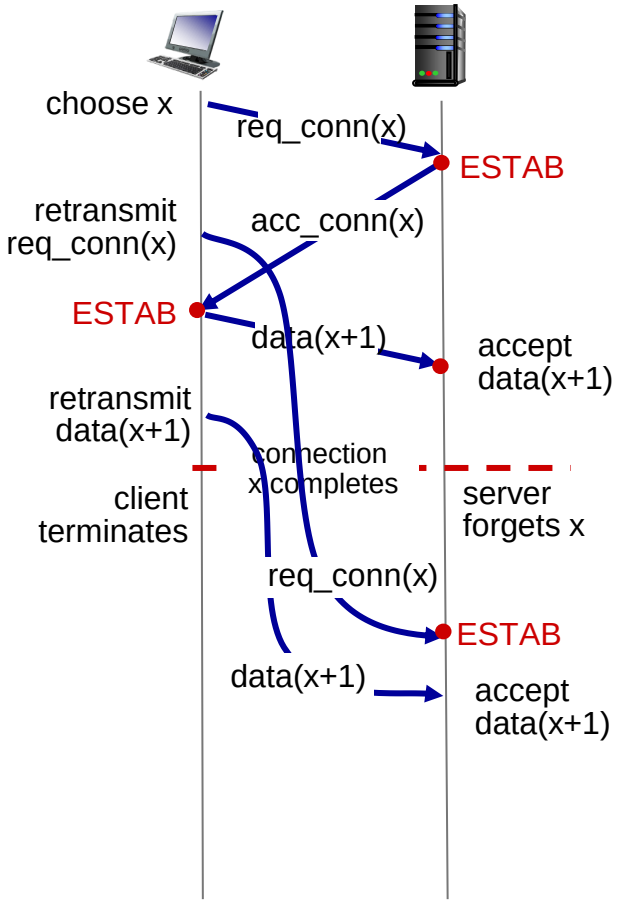


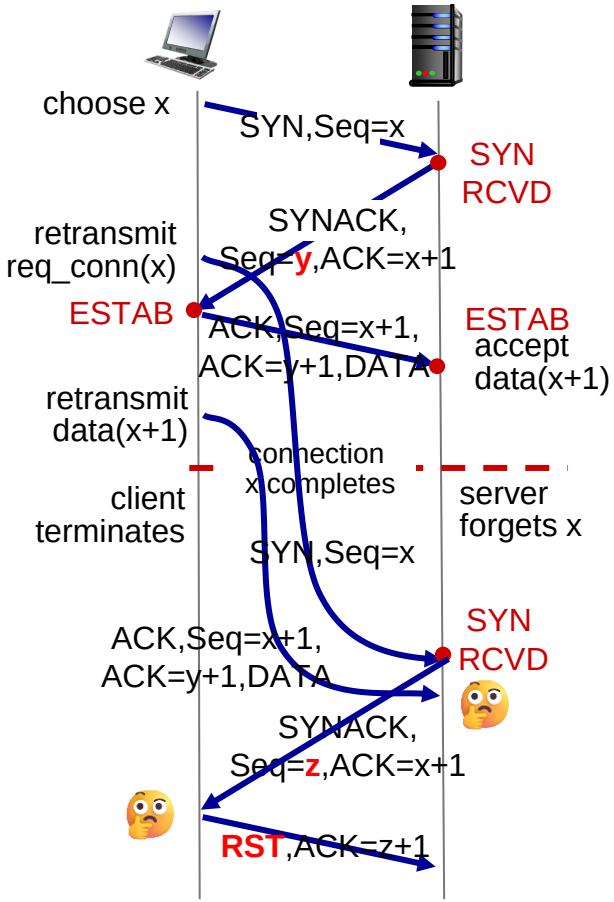
TCP 3-way Handshake

RST: Abort connection

- Receiver is confused
  - Unexpected sequence number
  - Port number or IP address do not match with any ongoing sockets

How it will solve the “duplicate data accept” problem?

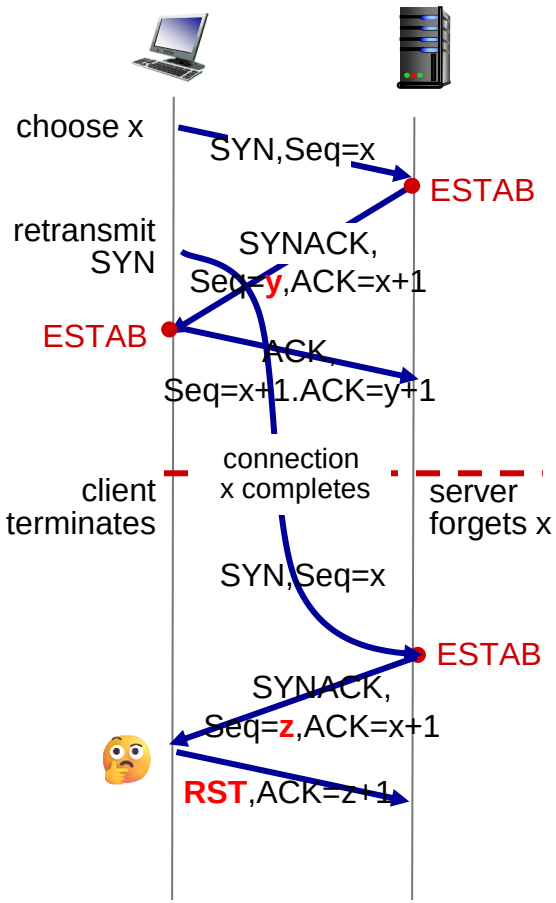


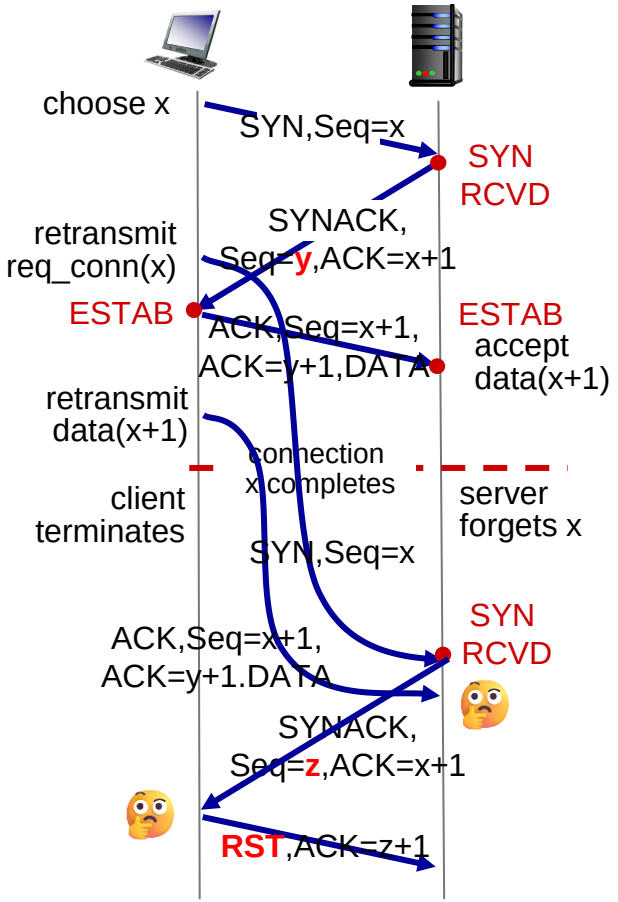


Initial Sequence Number

Initial sequence numbers are chosen randomly

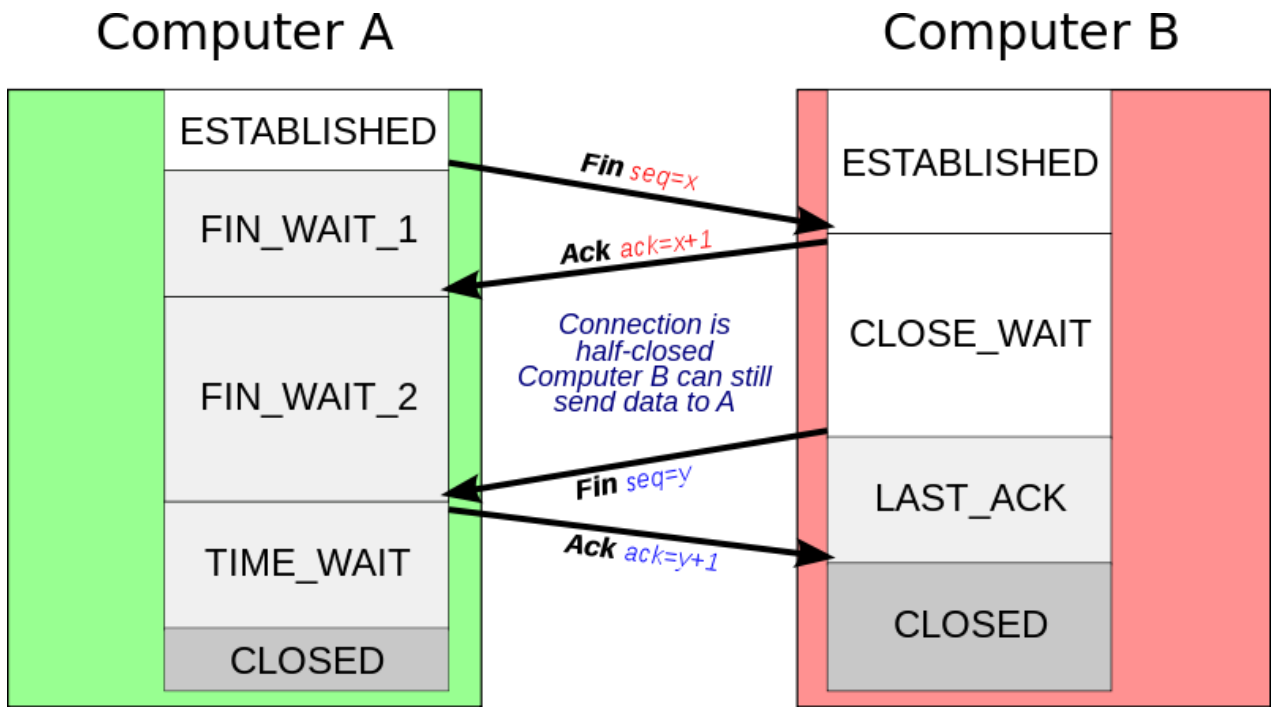
- Why not 0?
  - Segments of different connections can get mixed up
  - Security issues when sequence numbers are predictable





Closing a TCP Connection

- Client, server each close their side of connection
  - Send TCP segment with FIN bit = 1
- Respond to received FIN with ACK
  - On receiving FIN, ACK can be combined with own FIN
- All resources on the client side (including port number) are released



Src: https://commons.wikimedia.org/wiki/File:Tcp_close.svg

Closing a TCP Connection

- Symmetric connection termination
  - Both side should release separately
  - B may have some data to transmit

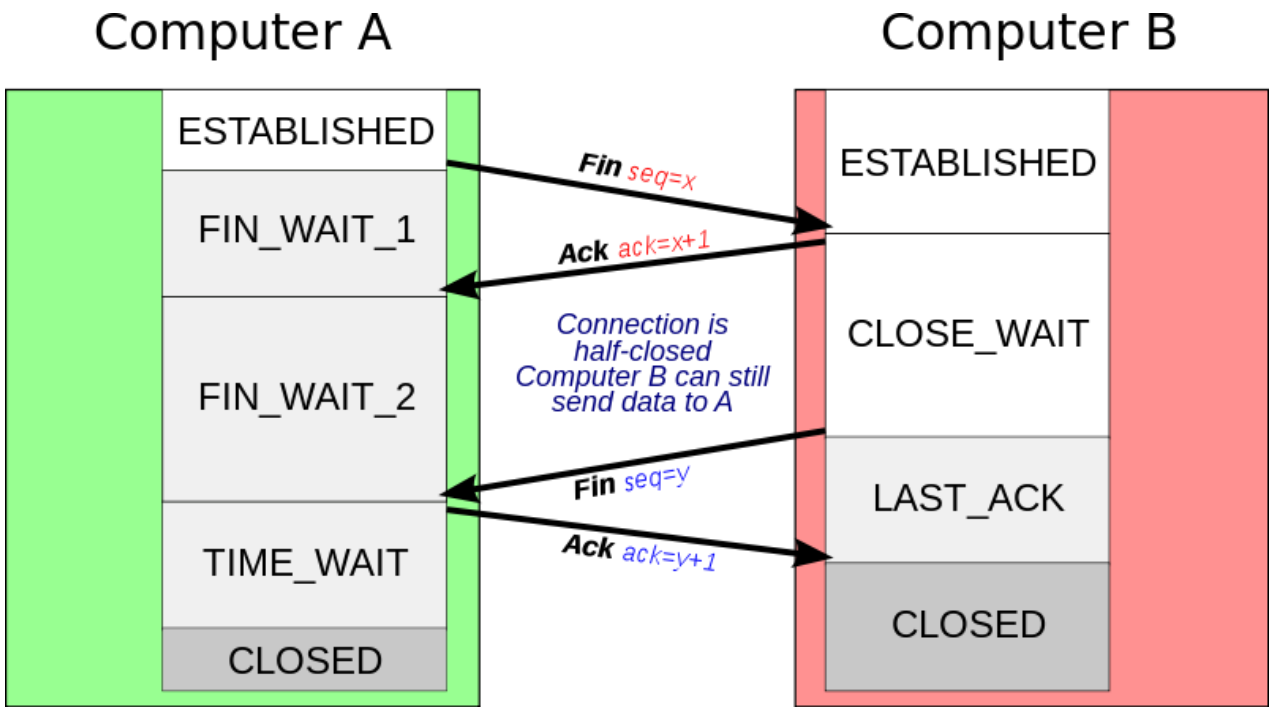
- TIME_WAIT varies from 30 secs - 2 mins
  - Typically equal to 2 x Max time an IP datagram might live in the Internet
  - Final ACK may be lost

Case-1:

- B may retransmit the FIN
- A can resend final ACK in case it is lost

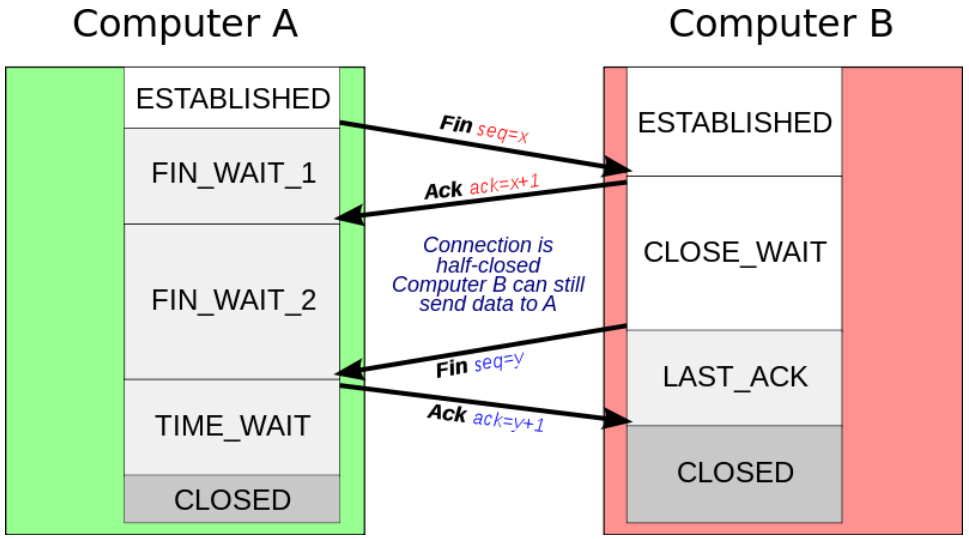
Case-2:

- The second FIN may get delayed
- Another pair of processes may open the connection (with same port numbers)
- The delayed FIN may initiate an unwanted termination

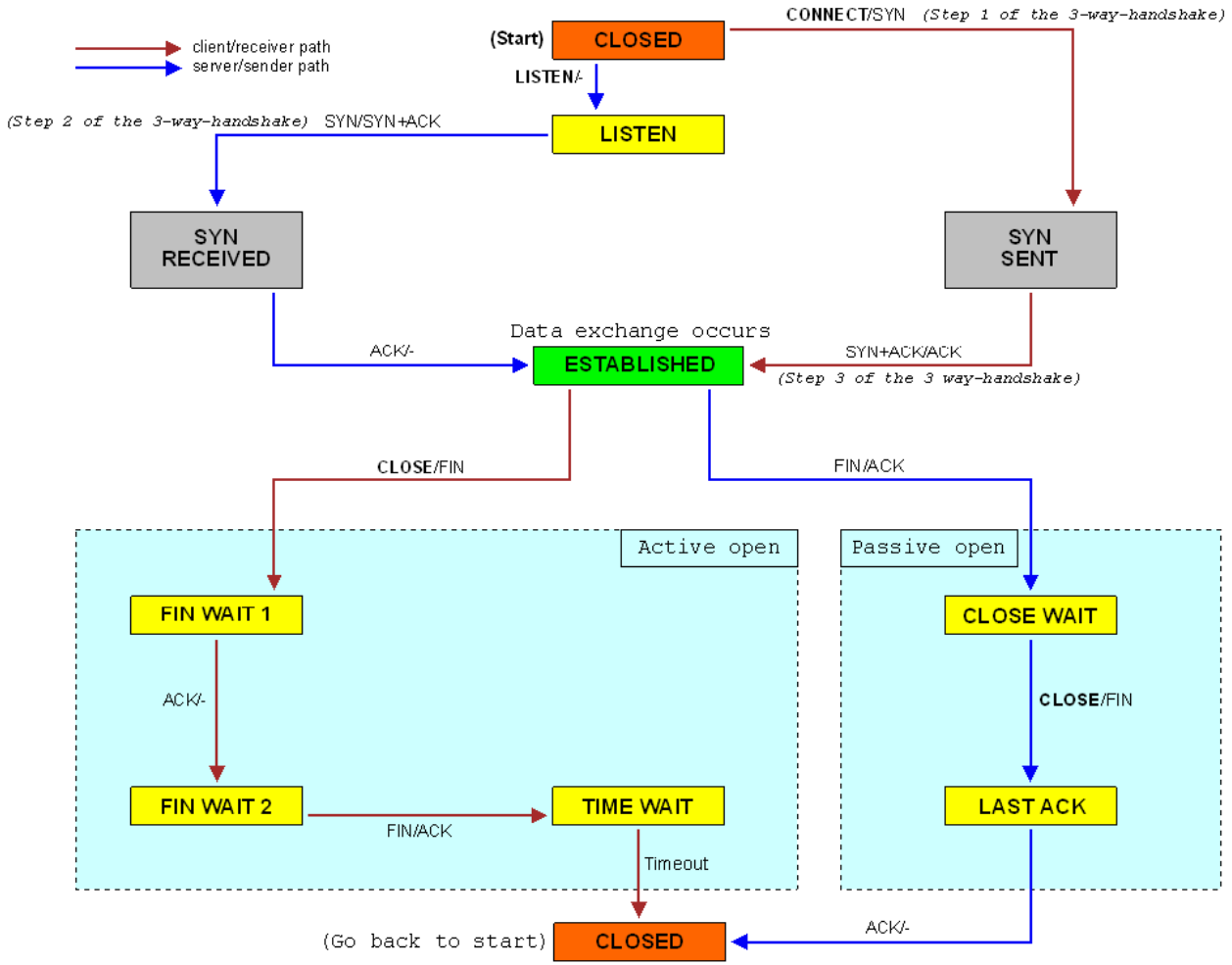


Src: https://commons.wikimedia.org/wiki/File:Tcp_close.svg

TCP State Diagram



Src: https://commons.wikimedia.org/wiki/File:Tcp_close.svg



Src: https://en.wikipedia.org/wiki/File:Tcp_state_diagram.png

Summary

- TCP connection management:
  - Connection setup → 3-way handshake
  - Connection termination → Symmetric release