TCP: Connection Management

Kameswari Chebrolu

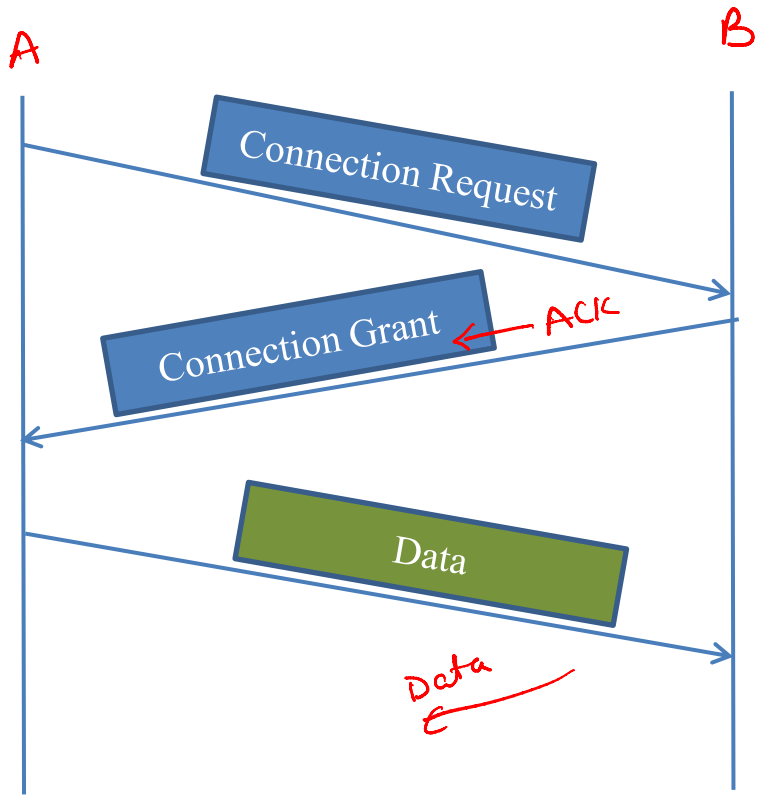
Background

- TCP is a connection oriented protocol
  - Processes can run on any type of machine in the Internet
- Connection establishment helps
  - Exchange and initiate state variables
    - MSS size, initial sequence number, ACK type
  - Allocate resources (buffer space)

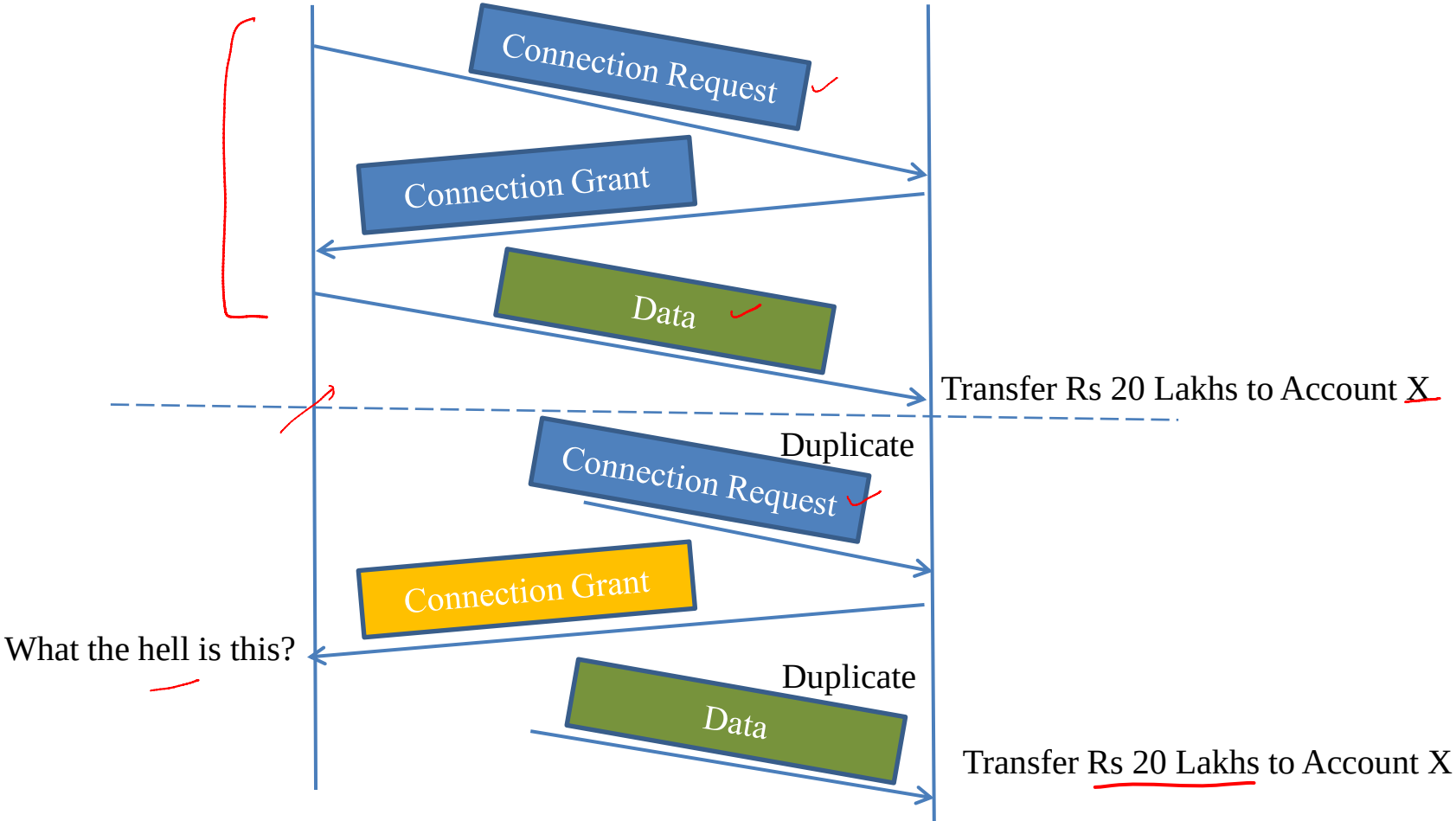
send Buffer
4KB – MB

receive Buffer
8KB

Connection Setup

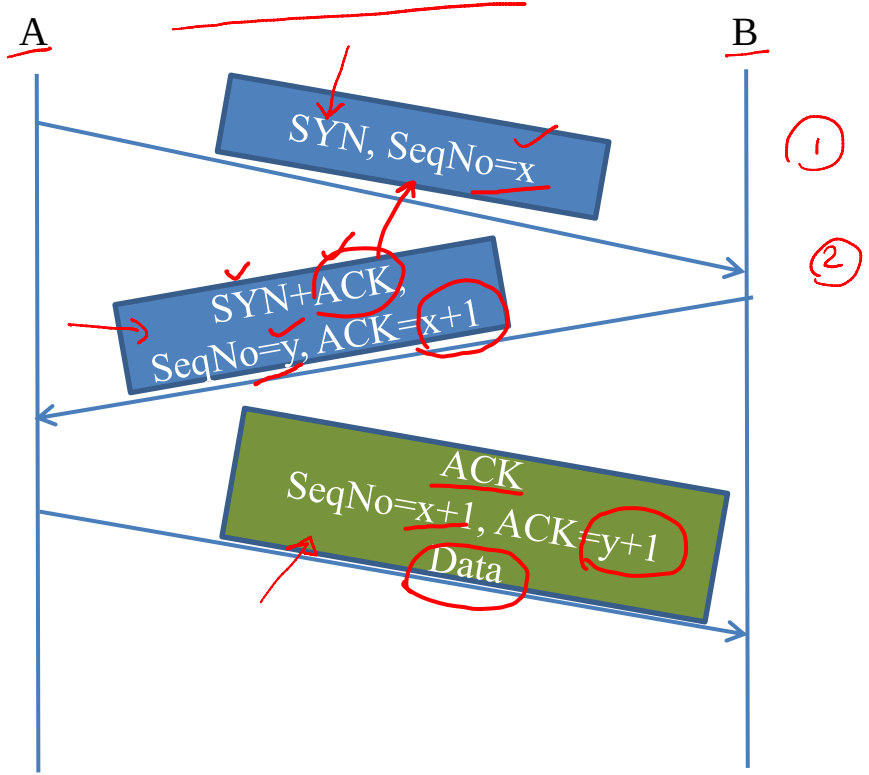


Problem

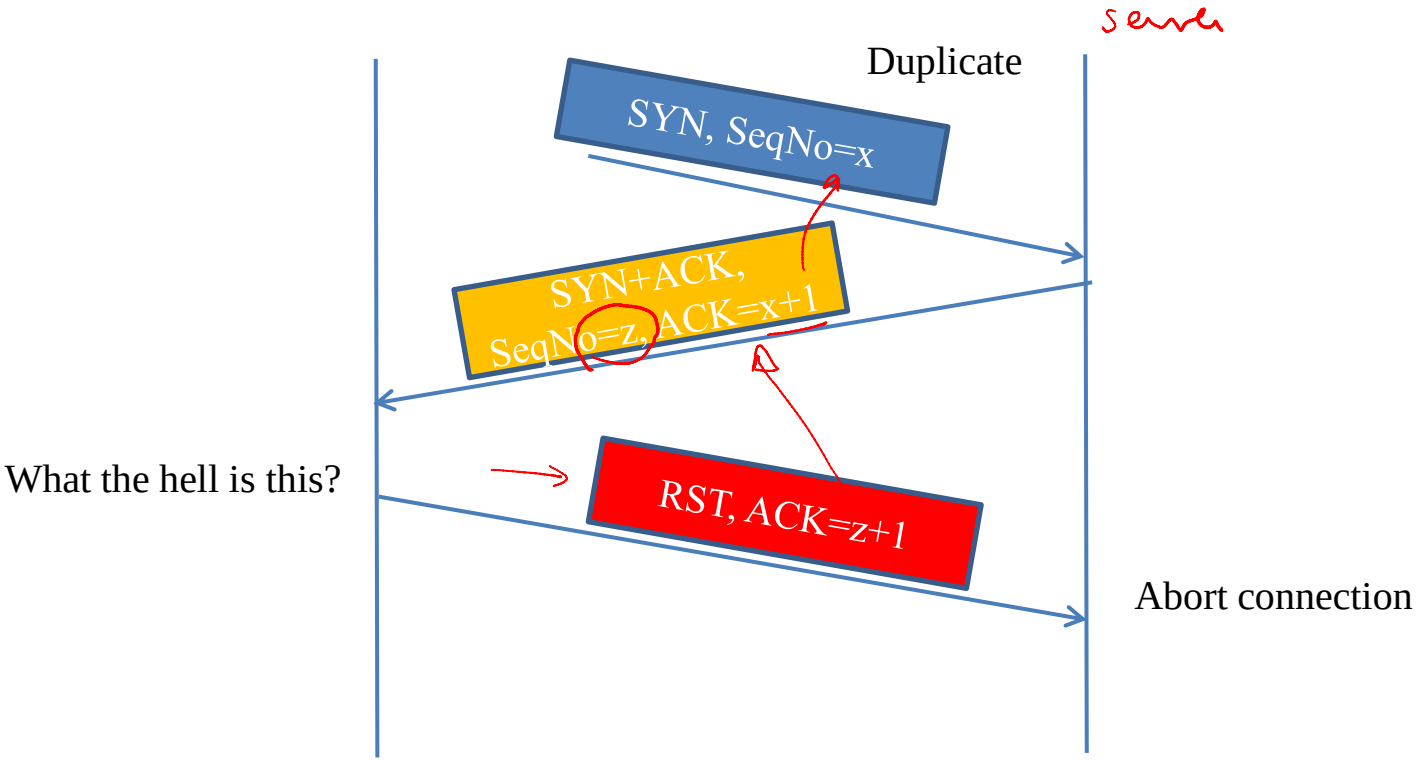

Connection Request
Connection Grant
Data
Transfer Rs 20 Lakhs to Account X
Duplicate
Connection Request
Connection Grant
What the hell is this?
Duplicate
Data
Transfer Rs 20 Lakhs to Account X

Solution

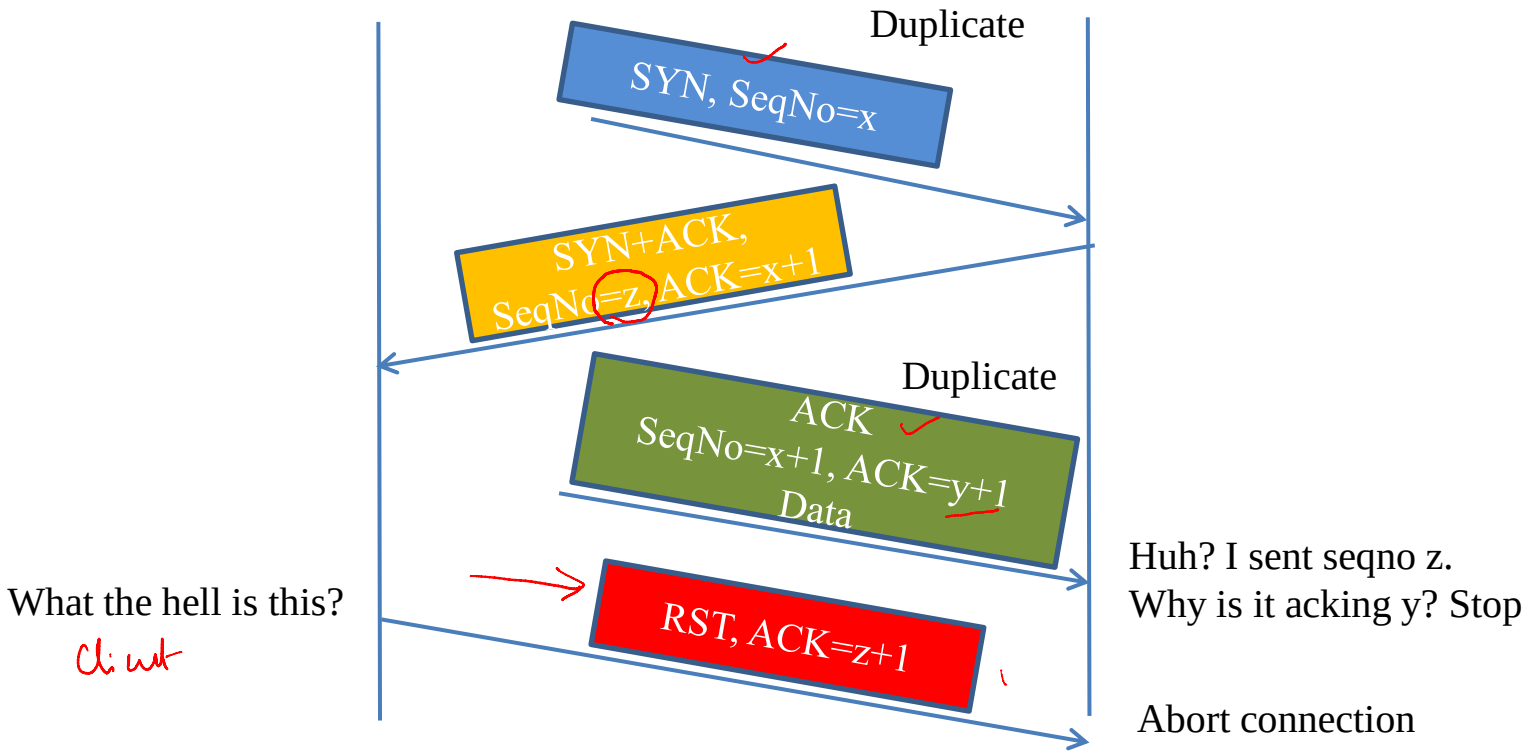
- TCP's famous three-way handshake (idea from Tomlinson)



Case-1


Duplicate
SYN, SeqNo=x
SYN+ACK,
SeqNo=z, ACK=x+1
What the hell is this?
RST, ACK=z+1
Abort connection

Case-2

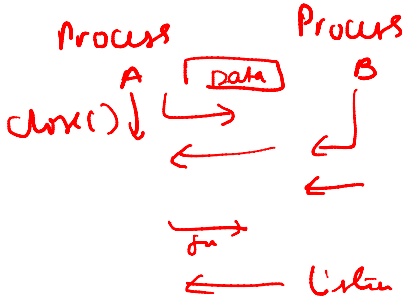

Duplicate
SYN, SeqNo=x
SYN+ACK,
SeqNo=z, ACK=x+1
Duplicate
ACK
SeqNo=x+1, ACK=y+1
Data
What the hell is this?
RST, ACK=z+1
Huh? I sent seqno z.
Why is it acking y? Stop
Abort connection

Initial Sequence Number (ISN)

- Why not start with Seqno zero?
- Segments from different connections can get mixed up
- Security risk when ISN's are predictable
- Original solution: Use a clock (e.g. increments every 4 microsec) to choose ISN
  - 32 bit sequence number wraps around in 4 hrs
- Current implementations use random ISN

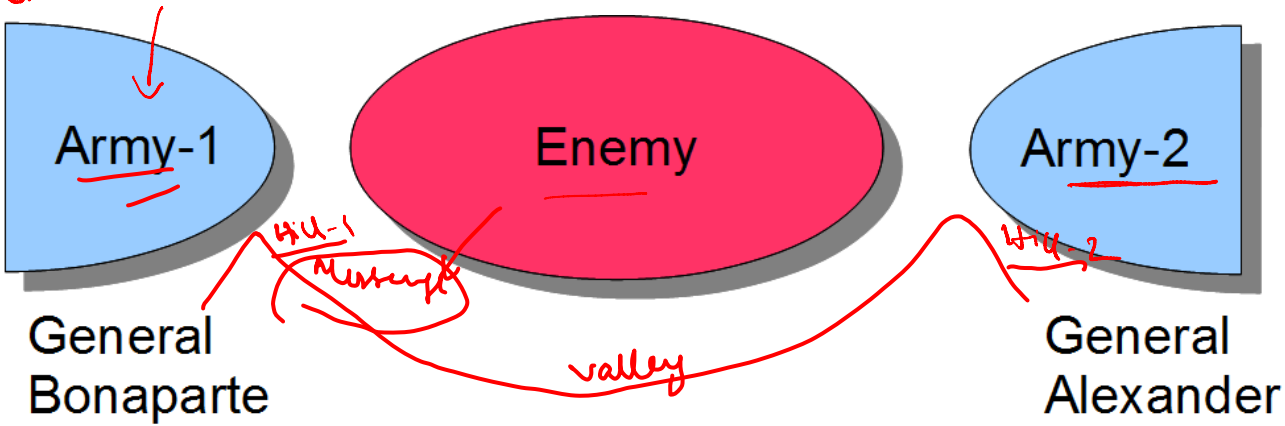
Connection Termination

- Asymmetric release (just hang-up) leads to loss of data
- Symmetric release
  - Treat connection as two separate unidirectional connections
  - Each side should be released separately



1. Let's attack on Sun @ 9am, Please ACK
2. OK · fine · Let's attack, Please ACK
3. OK · fine · ACK

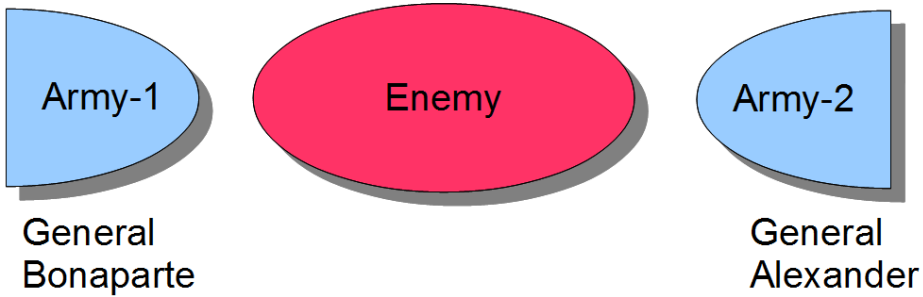
Two Army Problem



The attack will succeed *if and only if* both armies attack the enemy at the same time

What strategy to adopt?

Relevance



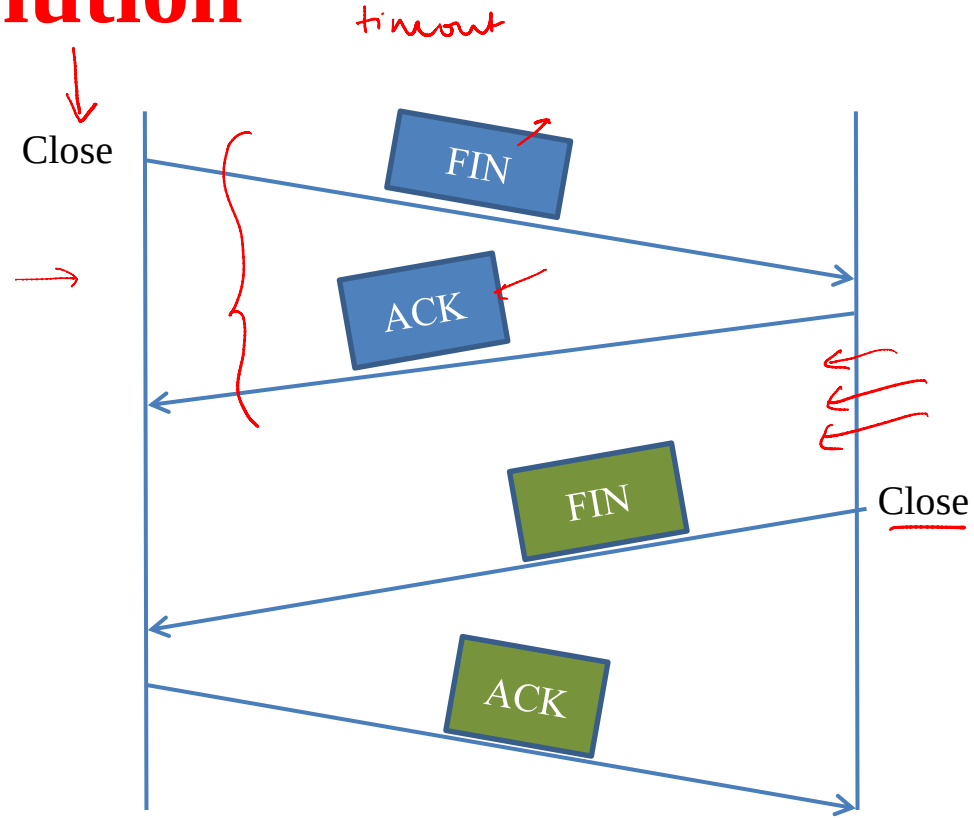
The attack will succeed *if and only if* both armies attack the enemy at the same time

What strategy to adopt?

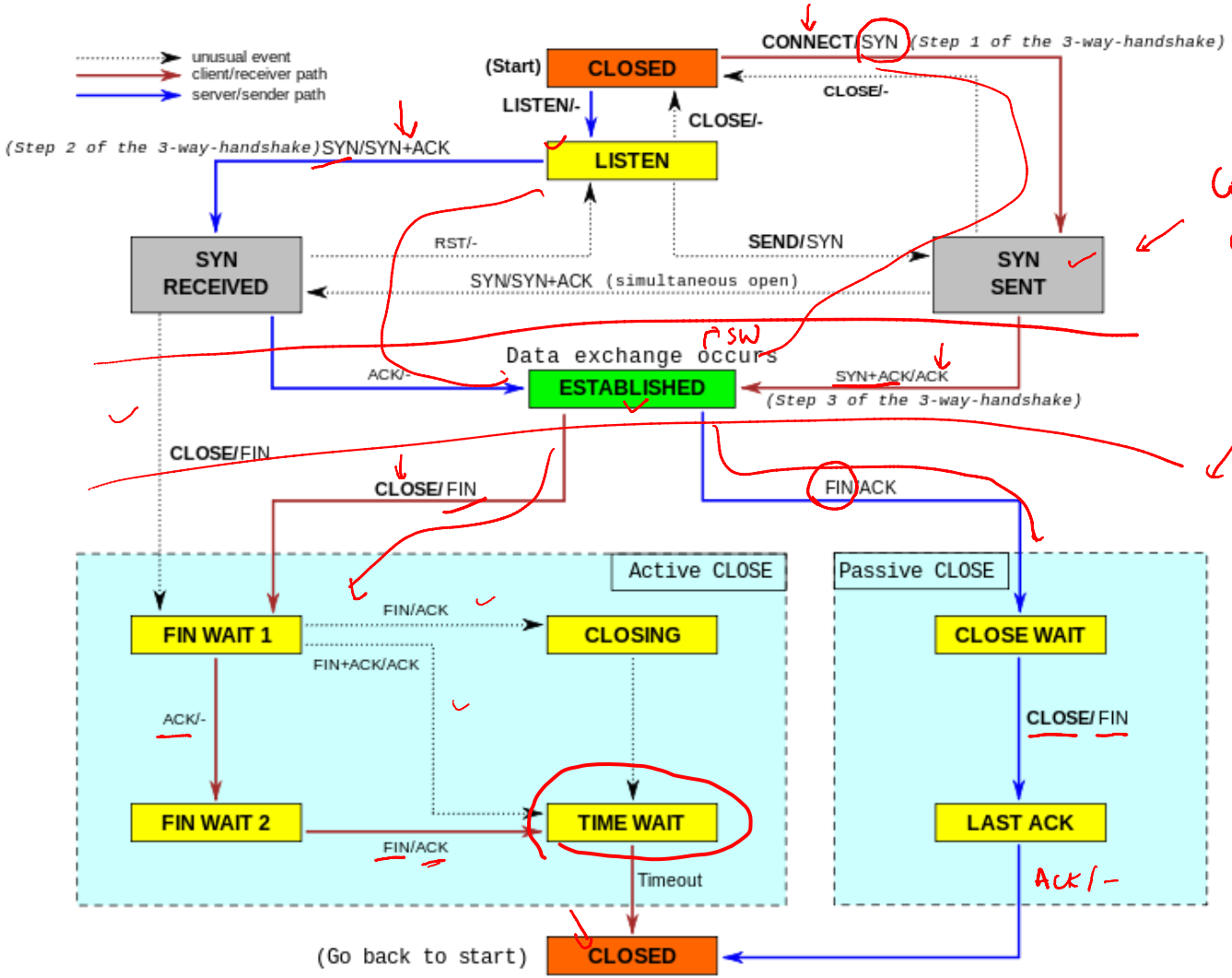
If neither side is ready to disconnect unless it is sure the other side is ready to disconnect, disconnect will never happen

Solution

- Follows simple two-way handshake
- Each side independently closes connection

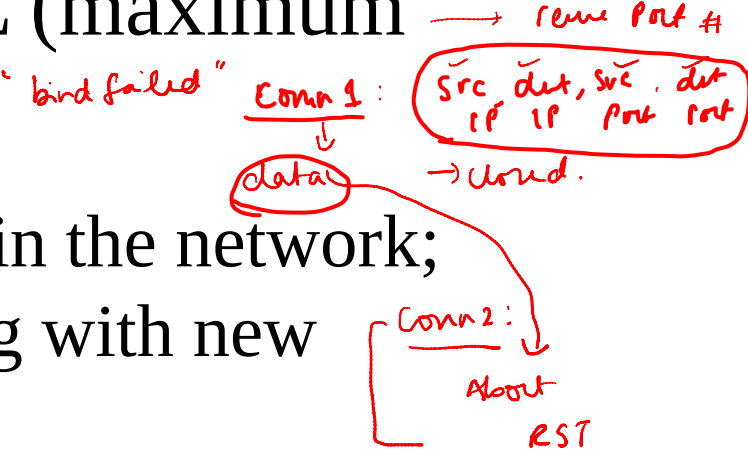


TCP State Diagram



Time-Wait State

- Wait in time-wait for 2*MSL (maximum segment lifetime)
  - Helps clear out older packets in the network; prevents them from interfering with new connection
  - Time spent in time-wait range from 30sec to 2 min



Summary

- TCP is a connection oriented protocol
- Connection management complicated by the fact that packets can get retransmitted, delayed, delivered out of order etc
- Connection establishment governed by 3-way handshake
- Connection termination is based on symmetric release and managed by 2-way handshake
- Ahead: Sliding window action in the established state