

TCP Connection Management

Created by

M Bateman, A Ruddle & C Allison

As part of the TCP View project

- Both sender and receiver must be ready before start of data transfer
 - Sender and receiver need to agree on parameters
 - Receive buffer size, initial sliding window variables
- Sender and receiver must agree when transfer is over
 - Both sides must discard state
- This is signalling
 - Setup/teardown state at the endpoints
 - Compare to ‘dialing’ in the telephone network

- Setup
 - 3-way handshake
- Transfer
 - Sliding window, data and ACKs in both directions
- Teardown
 - 4-way handshake
- Client-server model
 - Initiator (client) server
 - Listener (server) responds, provides service



TCP

Connection Creation

Active participant
(client)

Passive participant
(server)



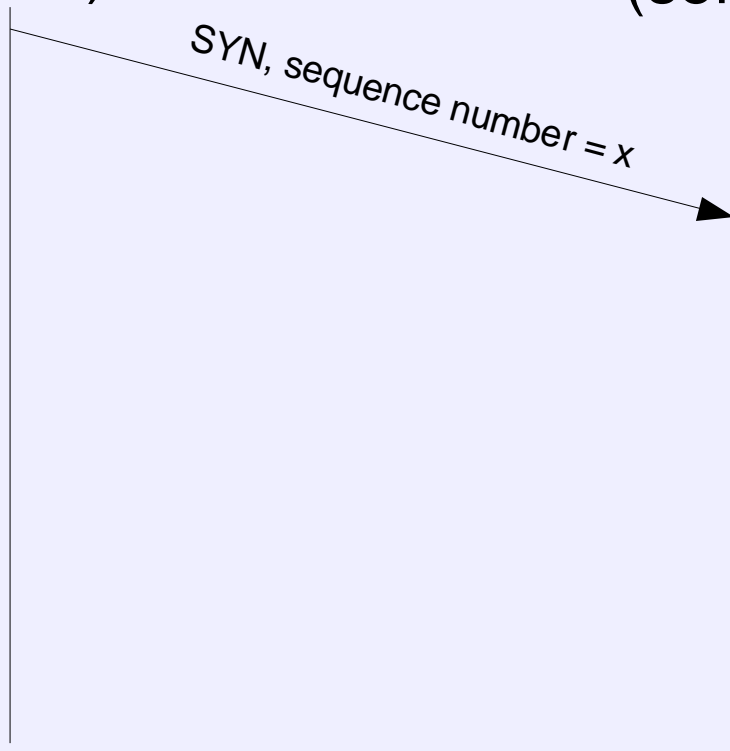
TCP

Connection Creation

Active participant
(client)

Passive participant
(server)

SYN, sequence number = x

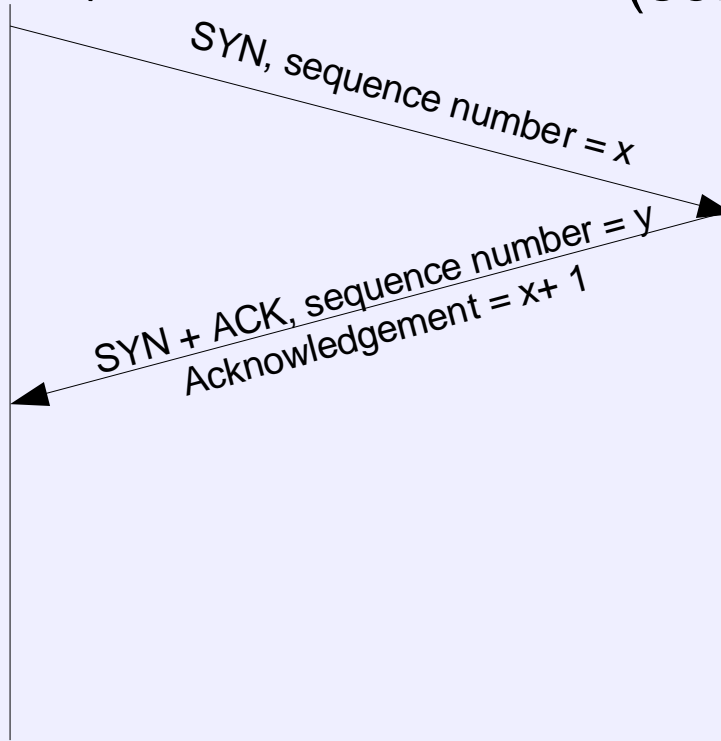


The diagram shows two vertical lines representing the lifelines of the client and server. An arrow points from the client's lifeline to the server's lifeline, labeled with the text 'SYN, sequence number = x'.

Connection Creation

Active participant
(client)

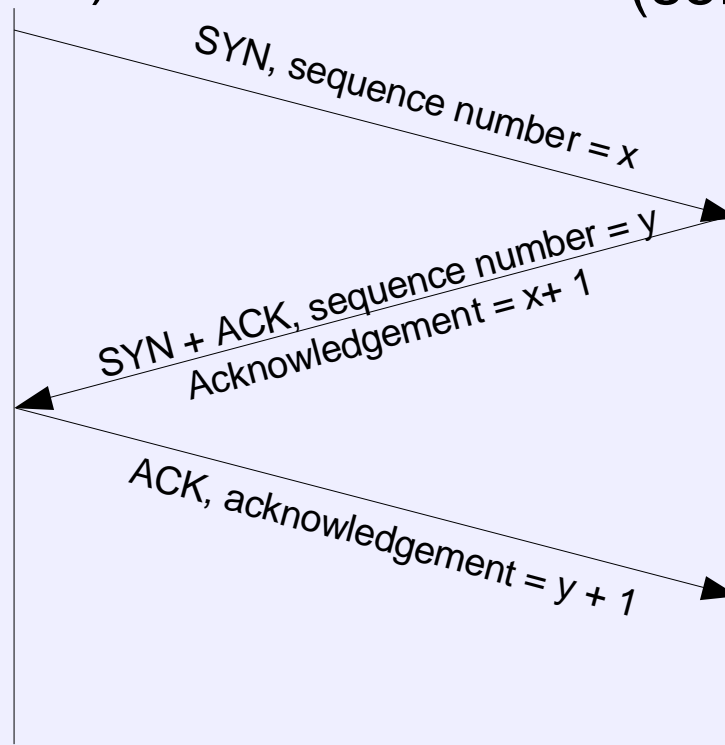
Passive participant
(server)



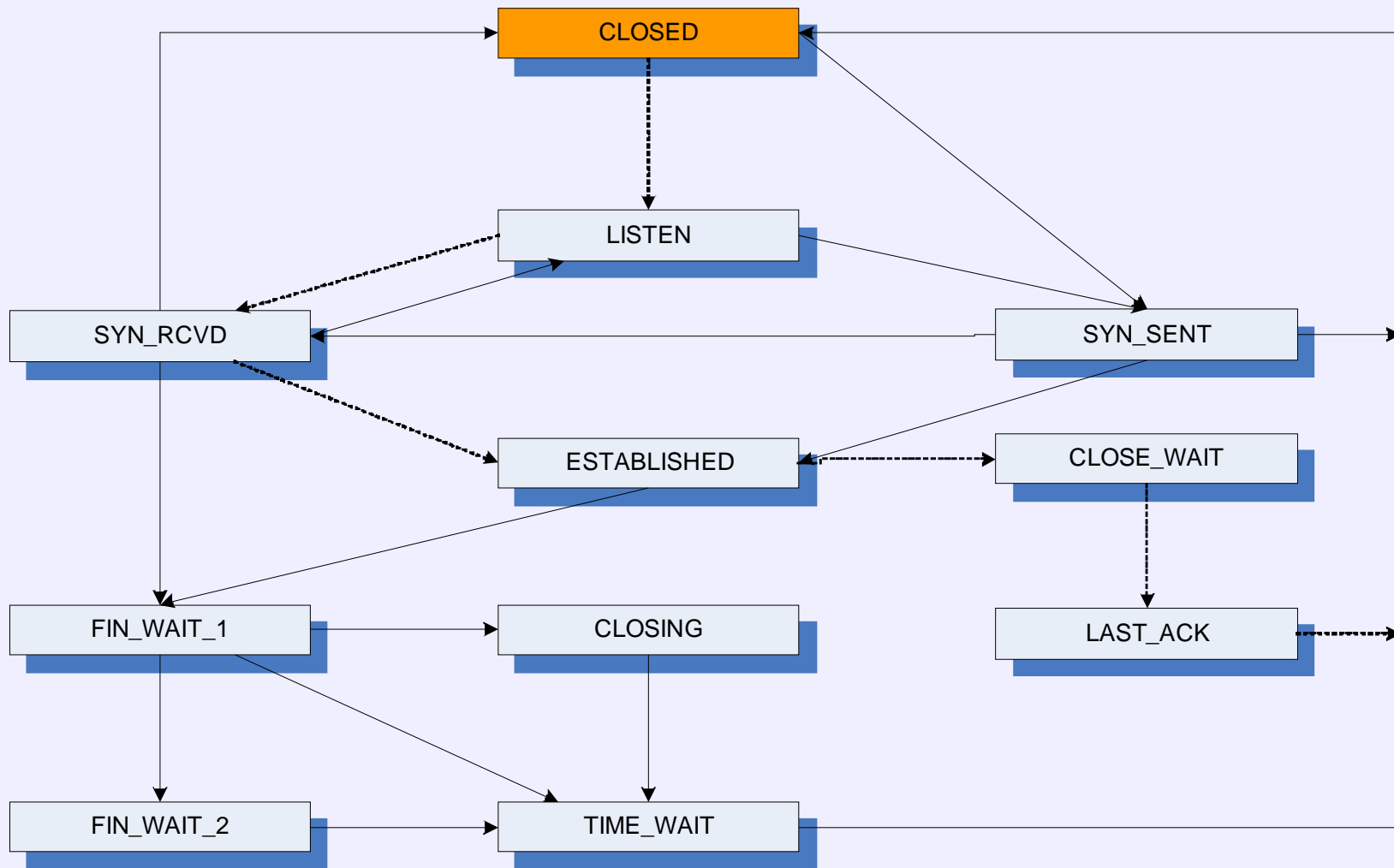
Connection Creation

Active participant
(client)

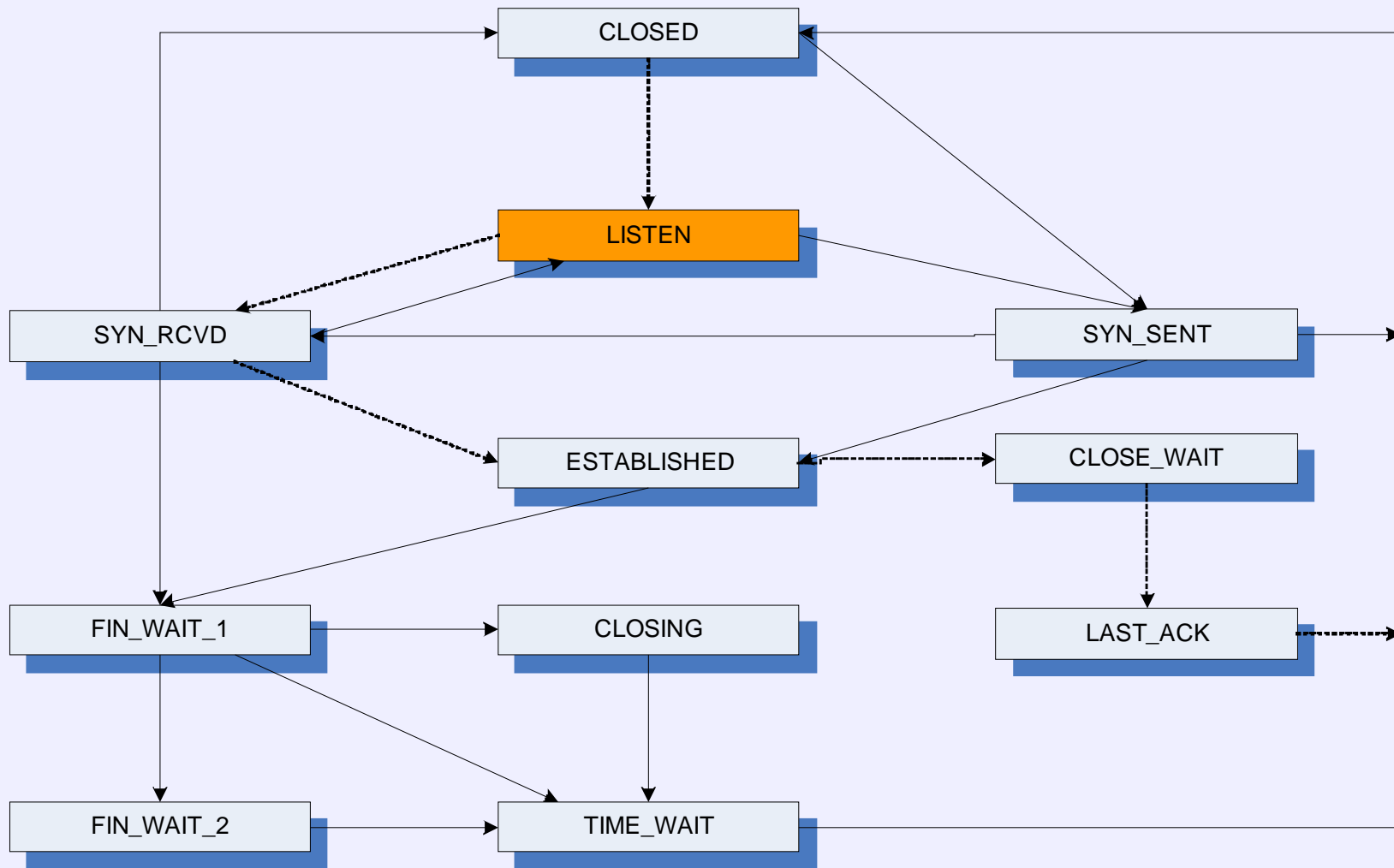
Passive participant
(server)



TCP



TCP



The logo features the letters 'TCP' in a bold, sans-serif font. The 'T' is red, while 'C' and 'P' are white. They are set against a dark grey rectangular background. To the right of the text, a bright yellow lightning bolt strikes downwards.

Data Transfer

- Connection is bi-directional
- ACKs can carry data



TCP

Connection Teardown

Active participant
(client)

Passive participant
(server)

TCP

Connection Teardown

Active participant
(client)

Passive participant
(server)

FIN

```
sequenceDiagram
    participant Client as Active participant (client)
    participant Server as Passive participant (server)
    Client->>Server: FIN
```

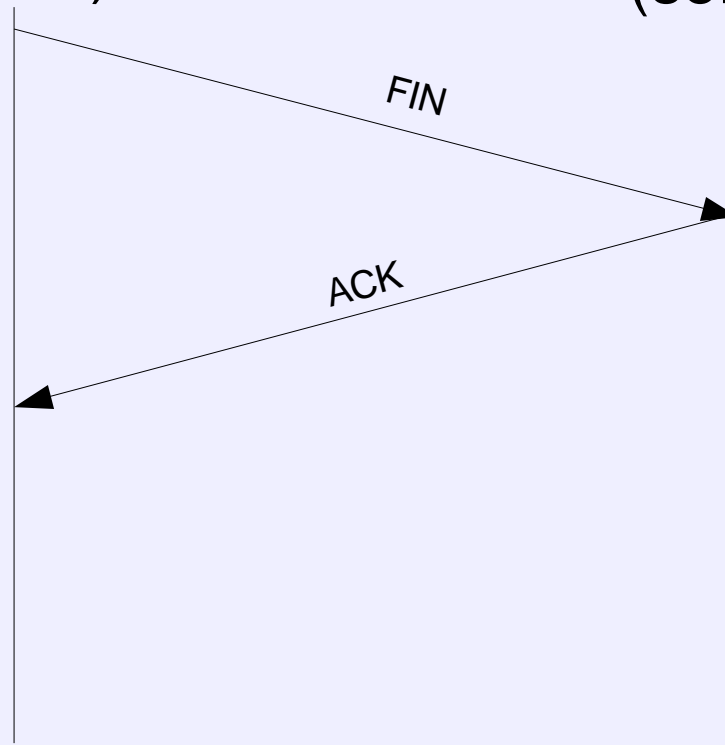
The diagram illustrates the first step of a TCP connection teardown. It features two vertical lines representing the lifelines of the participants: the 'Active participant (client)' on the left and the 'Passive participant (server)' on the right. A diagonal arrow points from the client's lifeline to the server's lifeline, labeled with the text 'FIN', indicating the transmission of a FIN segment to initiate the connection closure.

TCP

Connection Teardown

Active participant
(client)

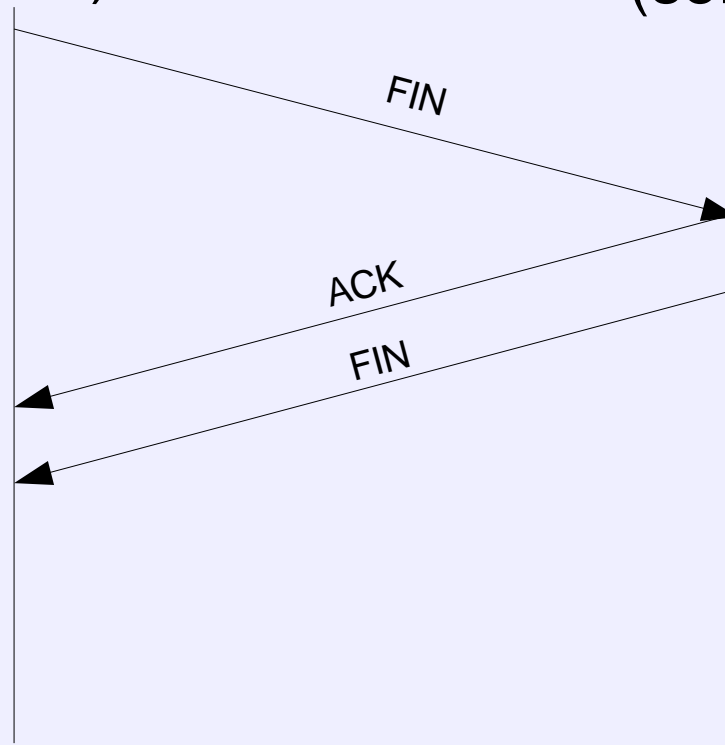
Passive participant
(server)



Connection Teardown

Active participant
(client)

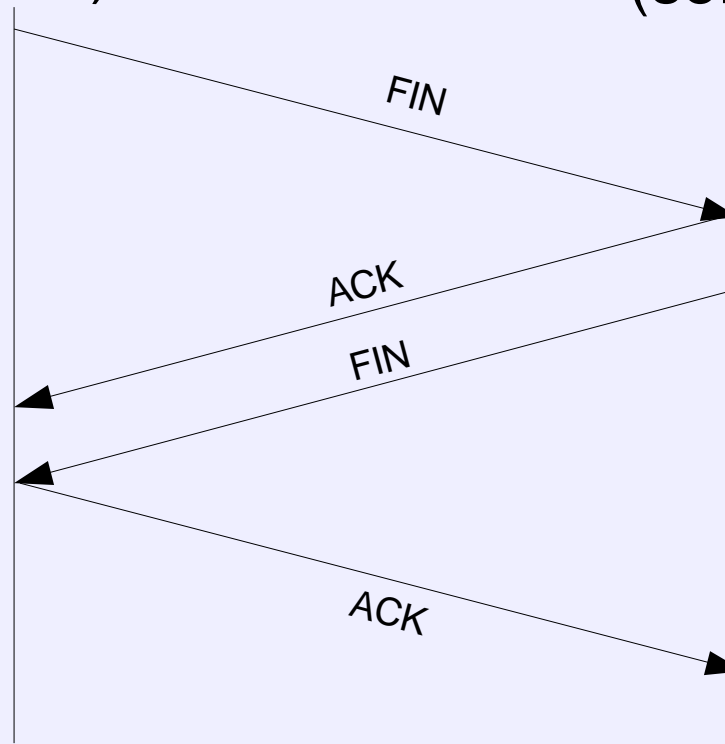
Passive participant
(server)



Connection Teardown

Active participant
(client)

Passive participant
(server)



The TIME_WAIT State

- We wait $2 * \text{maximum segment lifetime}$ (60 seconds) before completing the close
- Why?
 - ACK might have been lost so FIN would be resent
 - Could interfere with subsequent connections

‘Two-Army’ Problem

- Can we use messages and retries to synchronise two machines so they are guaranteed to do some operation at the same time?



TCP

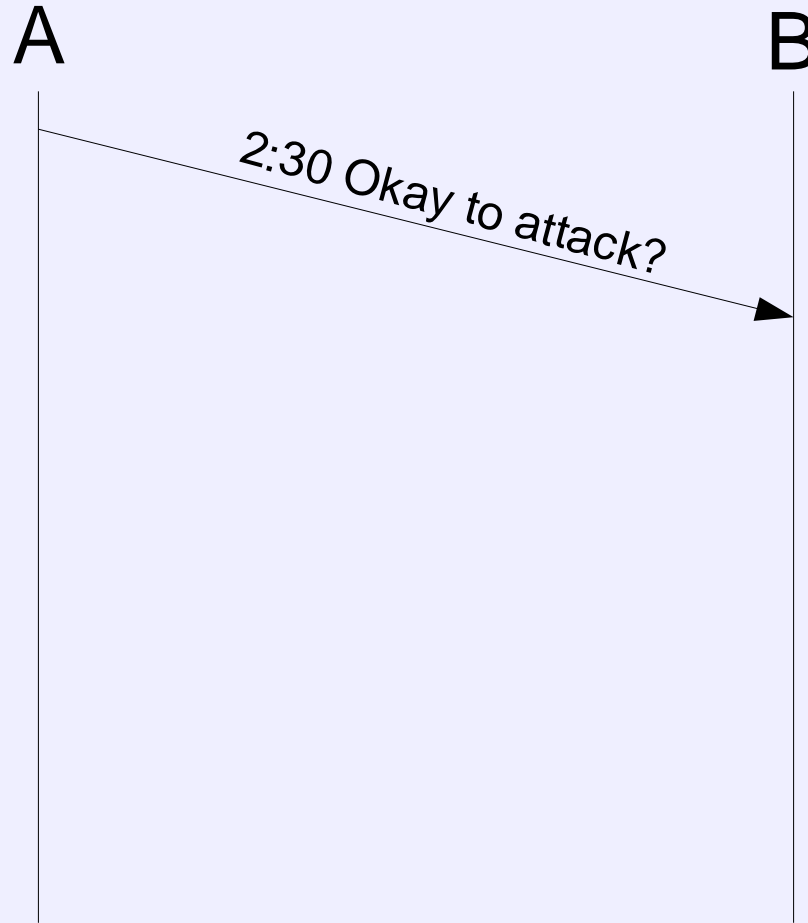
Two-Army Problem

A

B

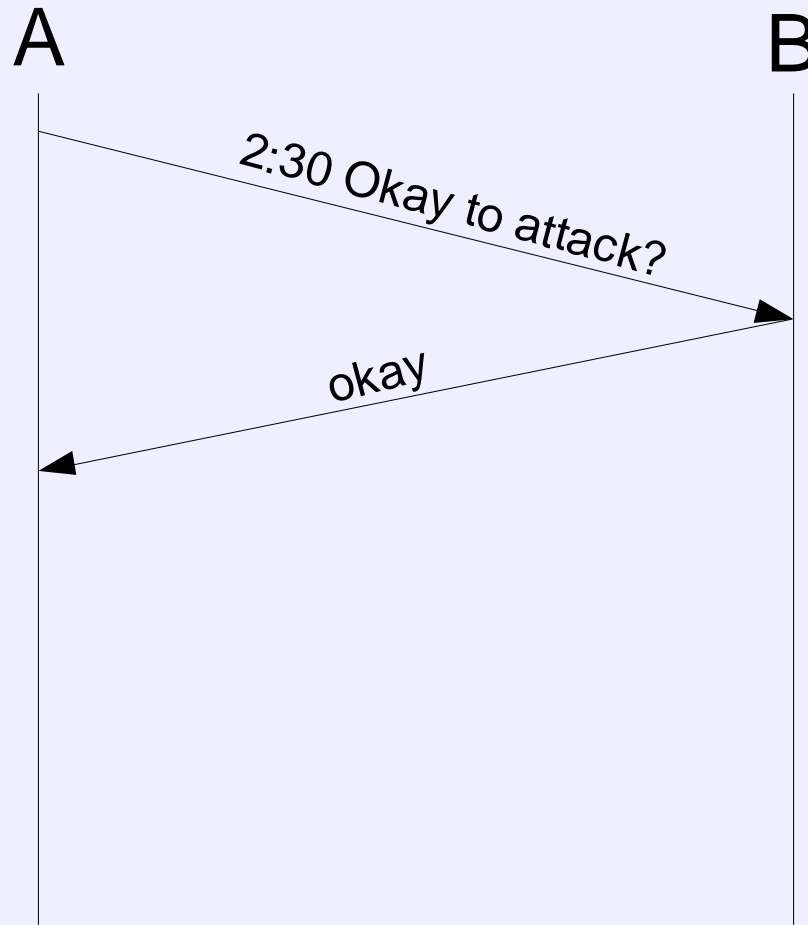
TCP

Two-Army Problem



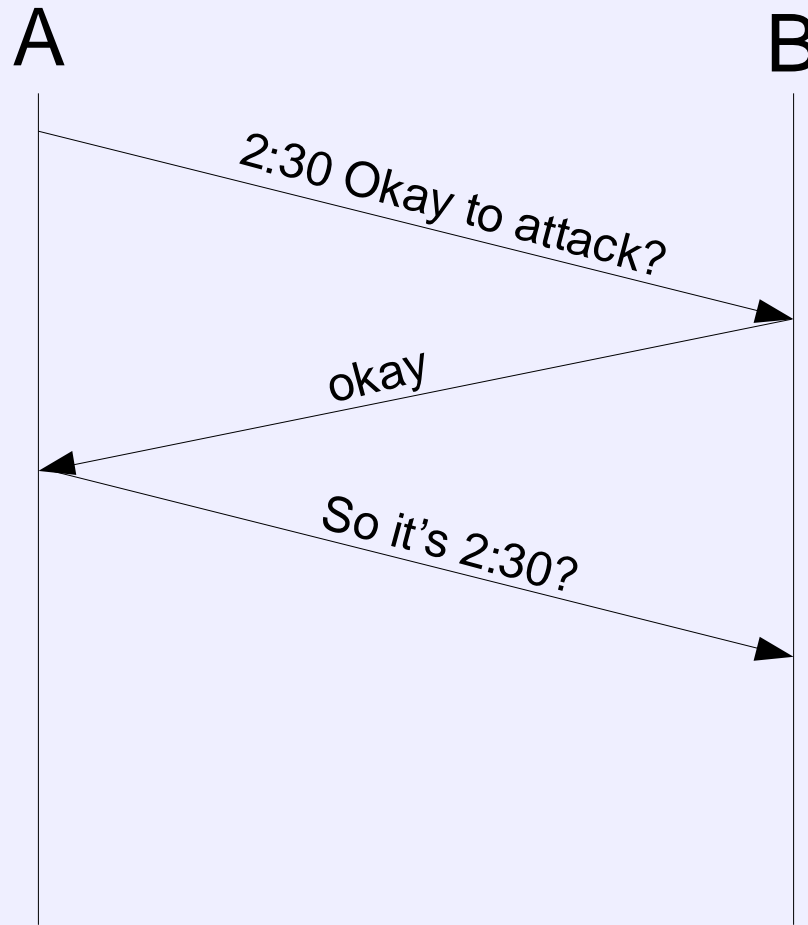
TCP

Two-Army Problem



TCP

Two-Army Problem



TCP

Two-Army Problem

