Introduction to Computer Networks

TCP Connection Management (II)

https://pages.cs.wisc.edu/~mgliu/CS640/F22/

Ming Liu
mgliu@cs.wisc.edu

Today

Last lecture

- How to setup the TCP connection?

Today

- How to tear down the TCP connection?

Announcements

- Lab4 is due 12/02/2022, 11:59 PM

Q: What is the goal of TCP connection management?

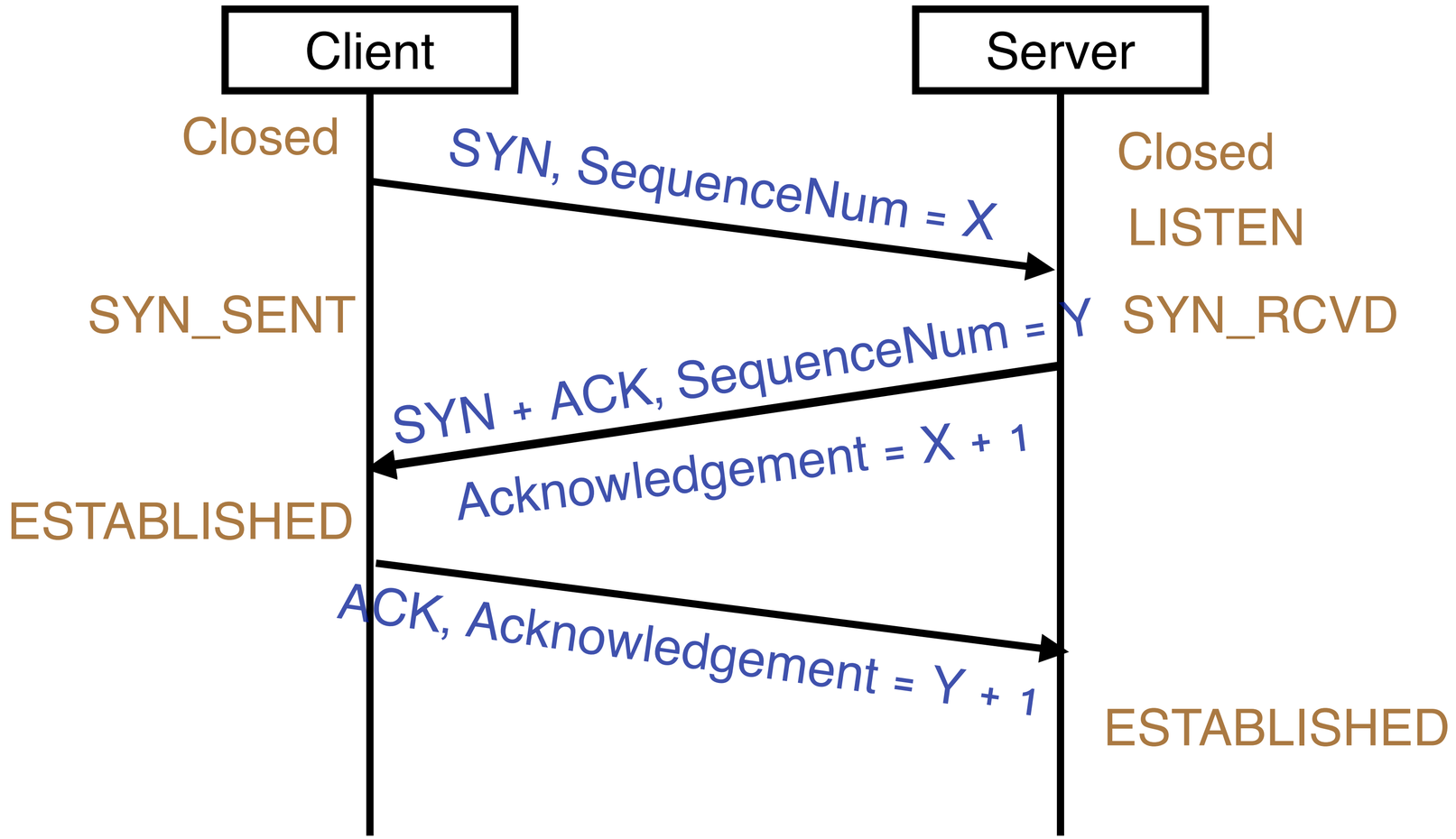
A: Dynamically create and destroy a full-duplex communication channel between a sender process and a receiver process for reliable byte stream exchange

On-demand communication

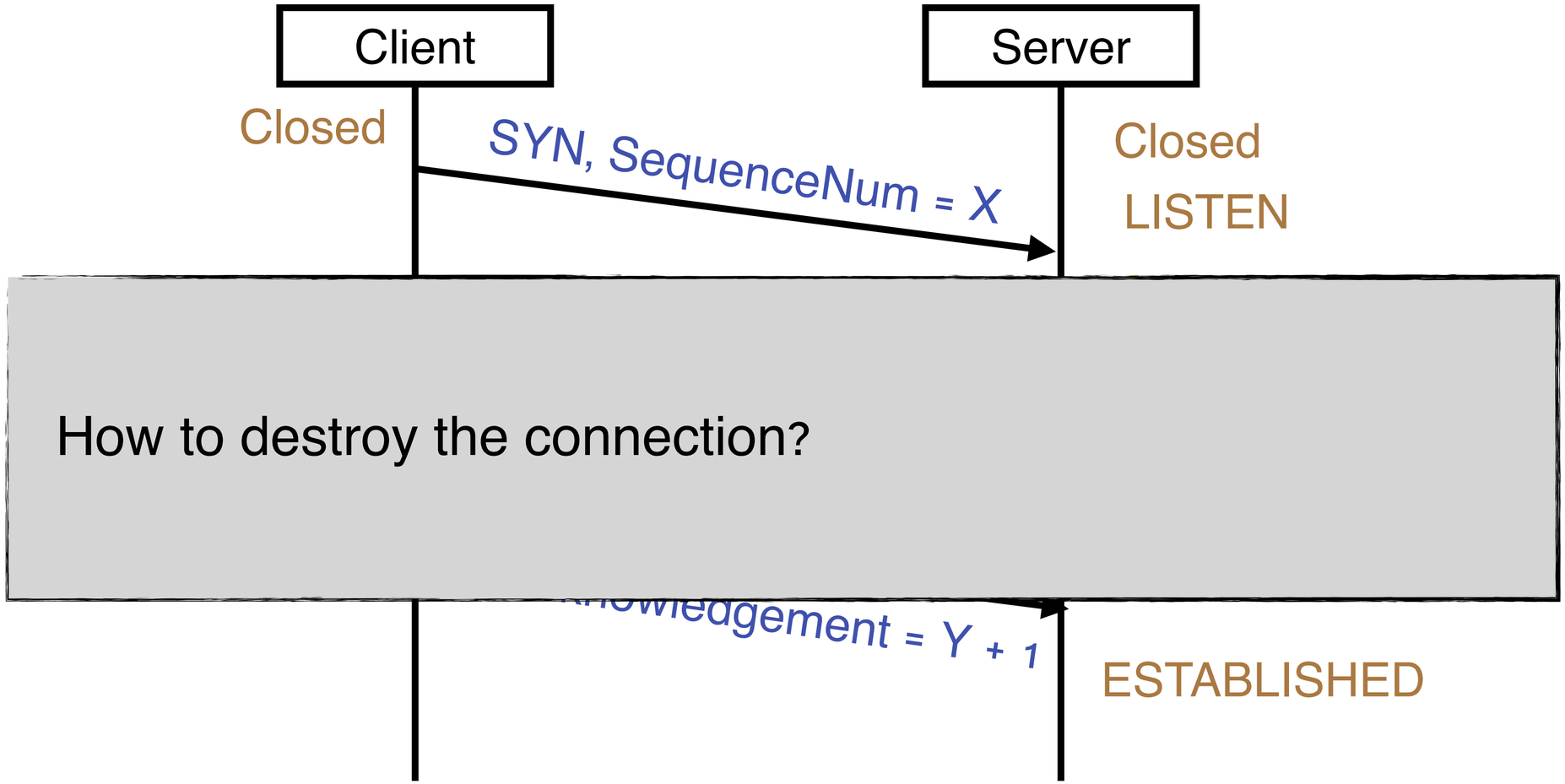
Client <—> Server

Client and server agree on the start of byte steams for two directions

TCP Connection Establishment Summary



TCP Connection Establishment Summary



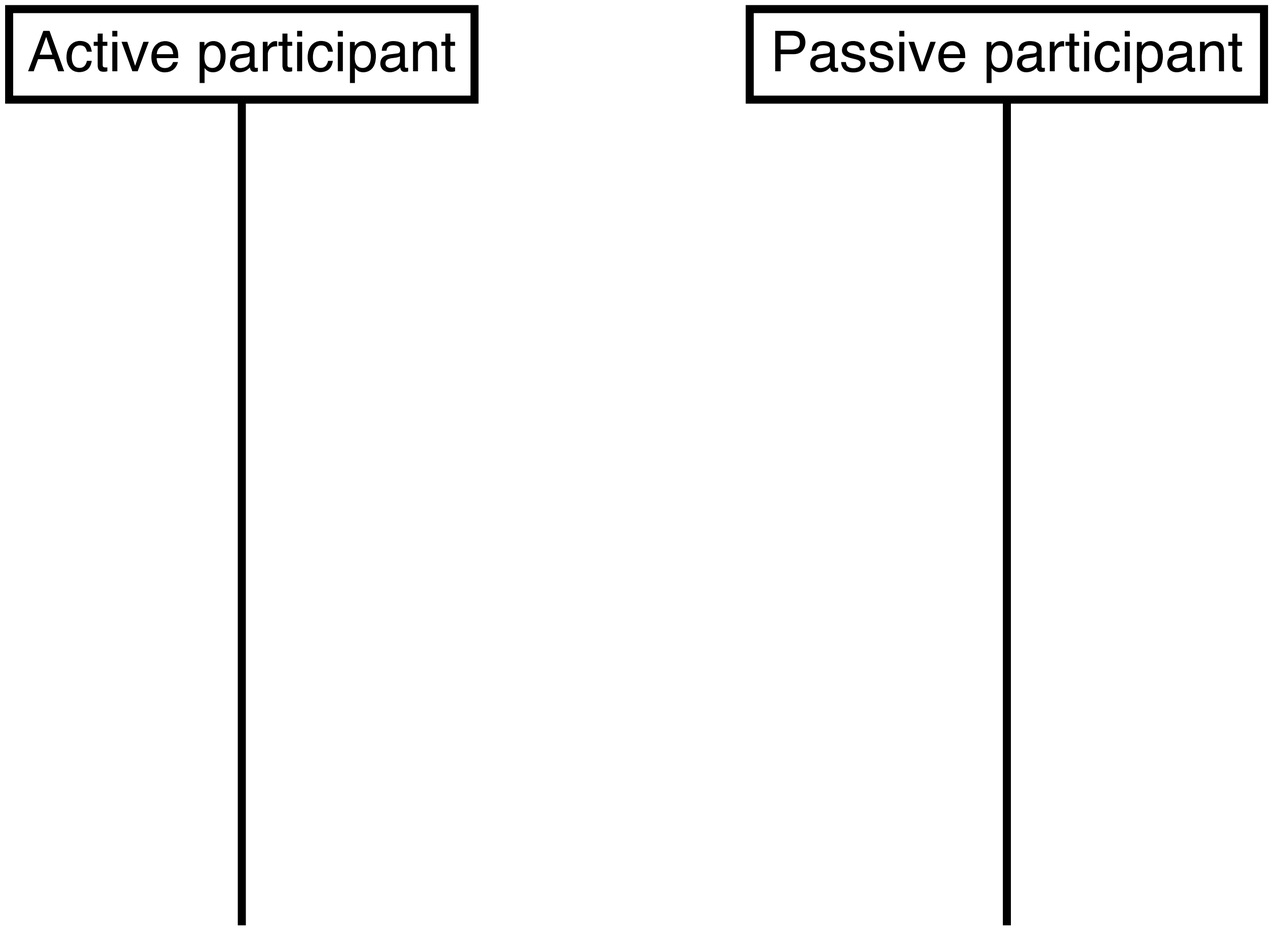
Connection Termination

Three cases:

- Case #1: One-side closes first
- Case #2: Both sides close simultaneously
- Case #3: Both sides close simultaneously (special)

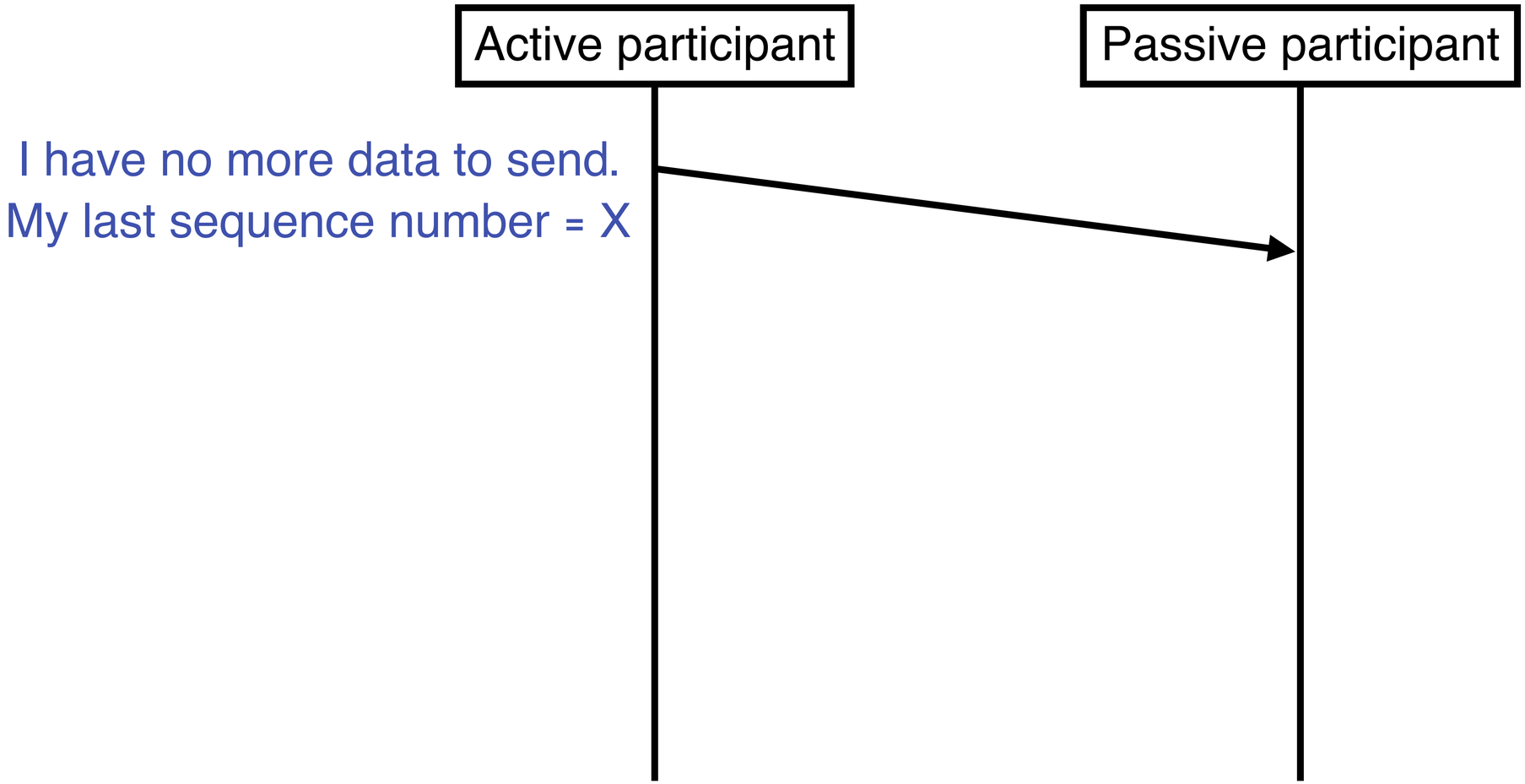
Case 1: One-side Closes First

4-way handshake



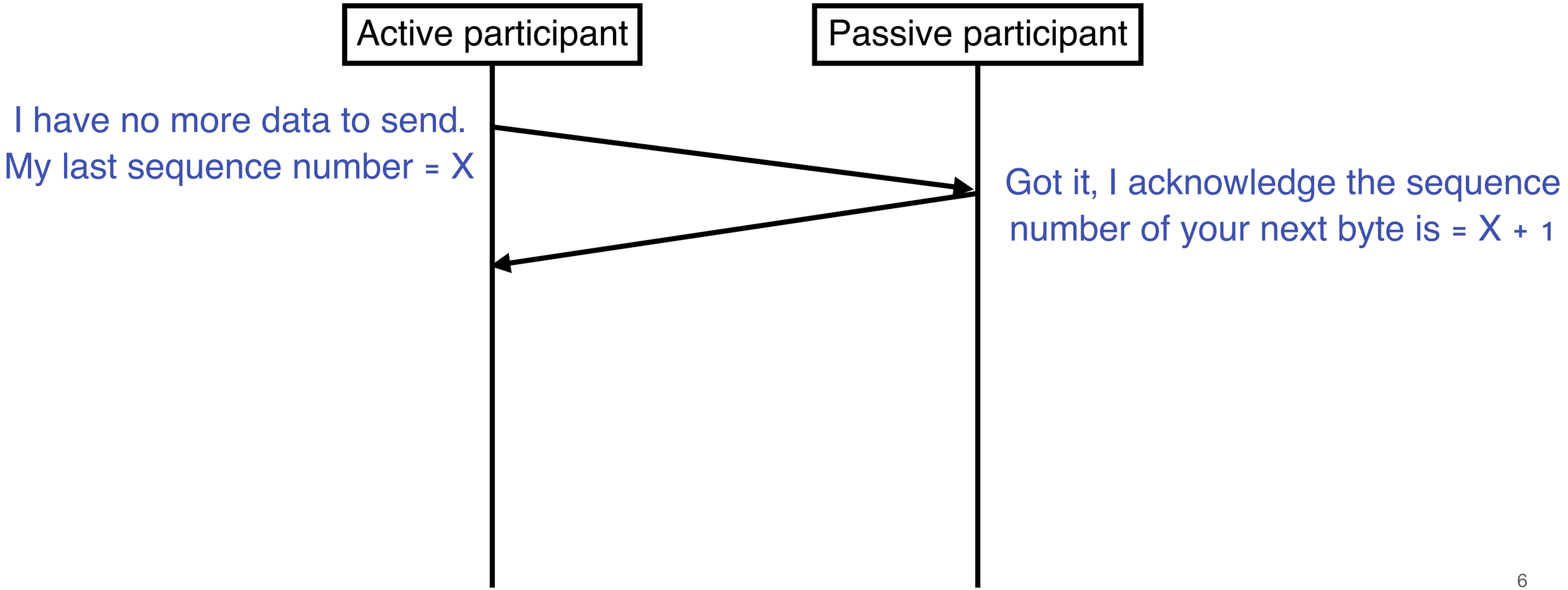
Case 1: One-side Closes First

4-way handshake



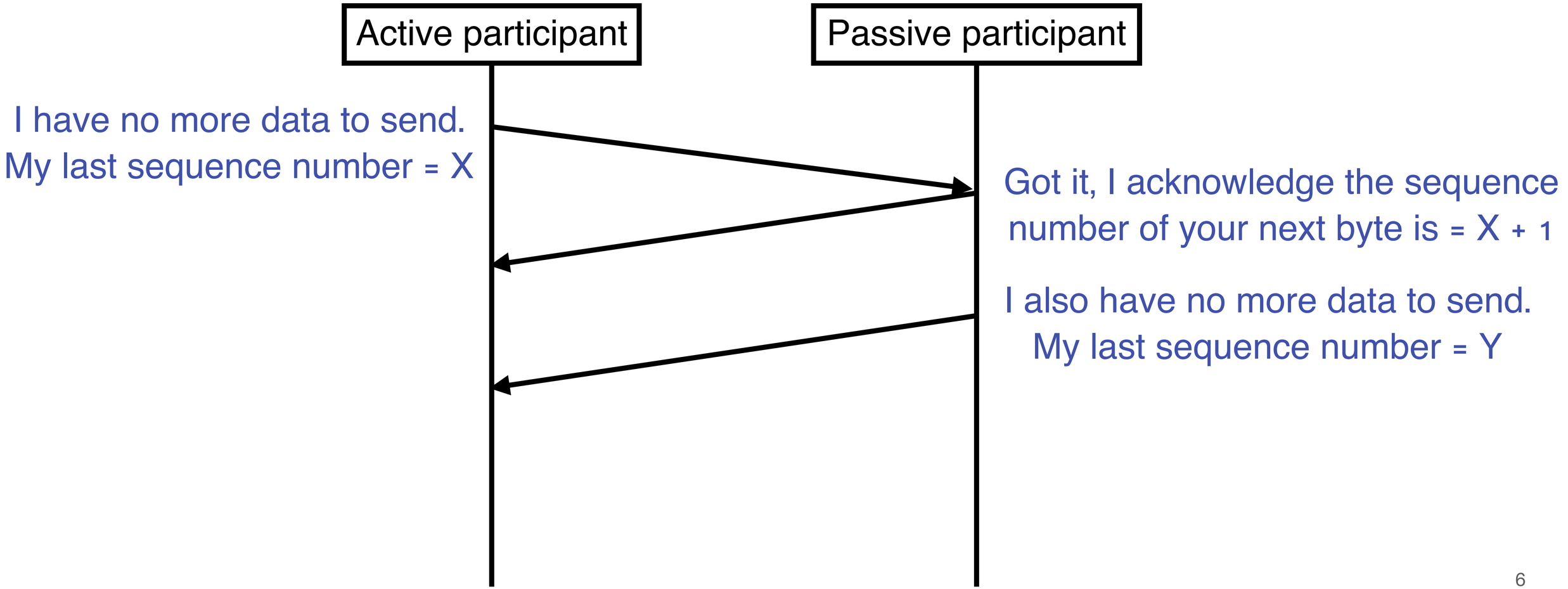
Case 1: One-side Closes First

4-way handshake



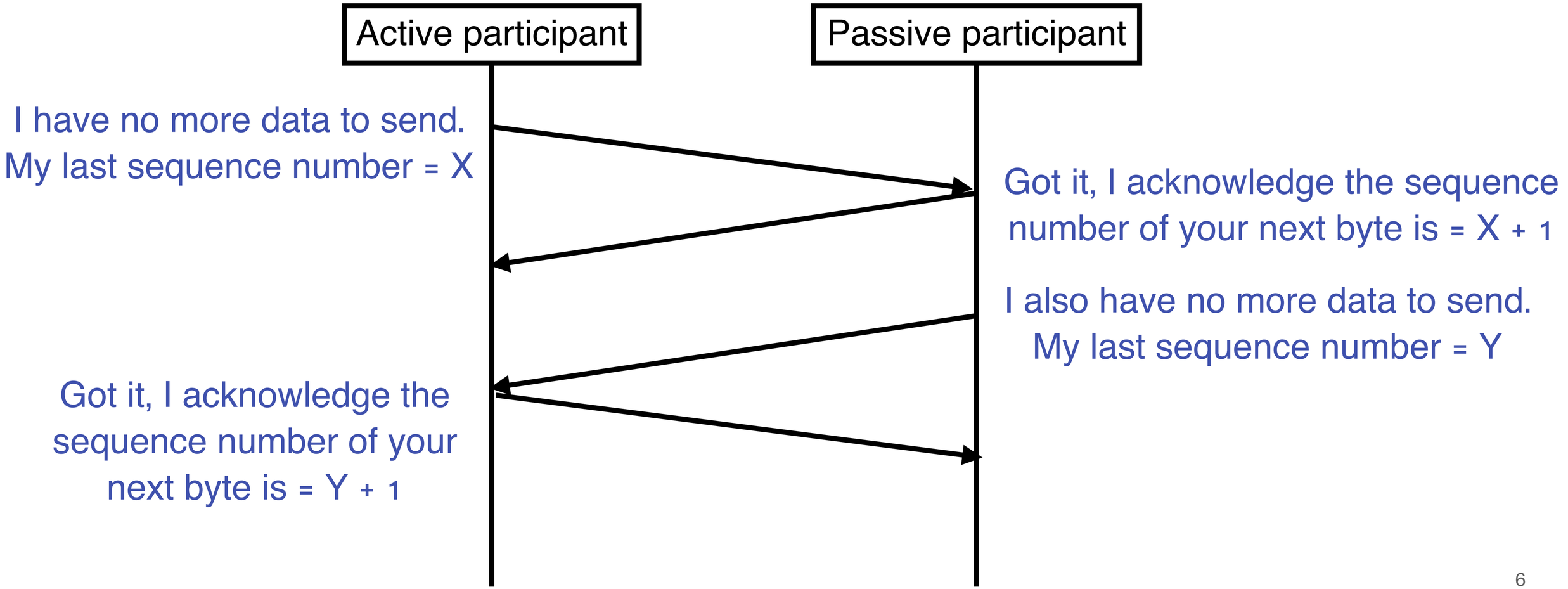
Case 1: One-side Closes First

4-way handshake



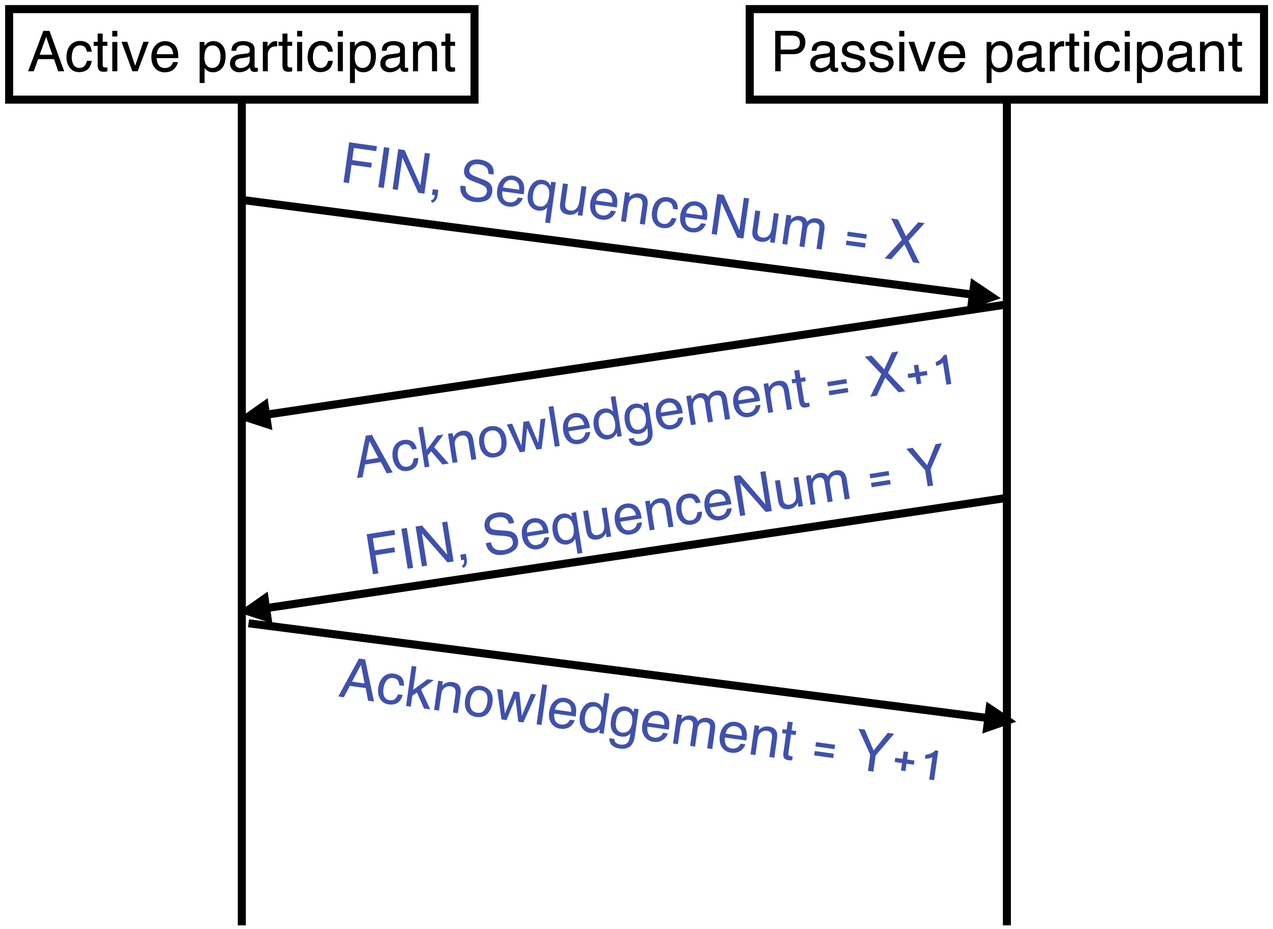
Case 1: One-side Closes First

4-way handshake



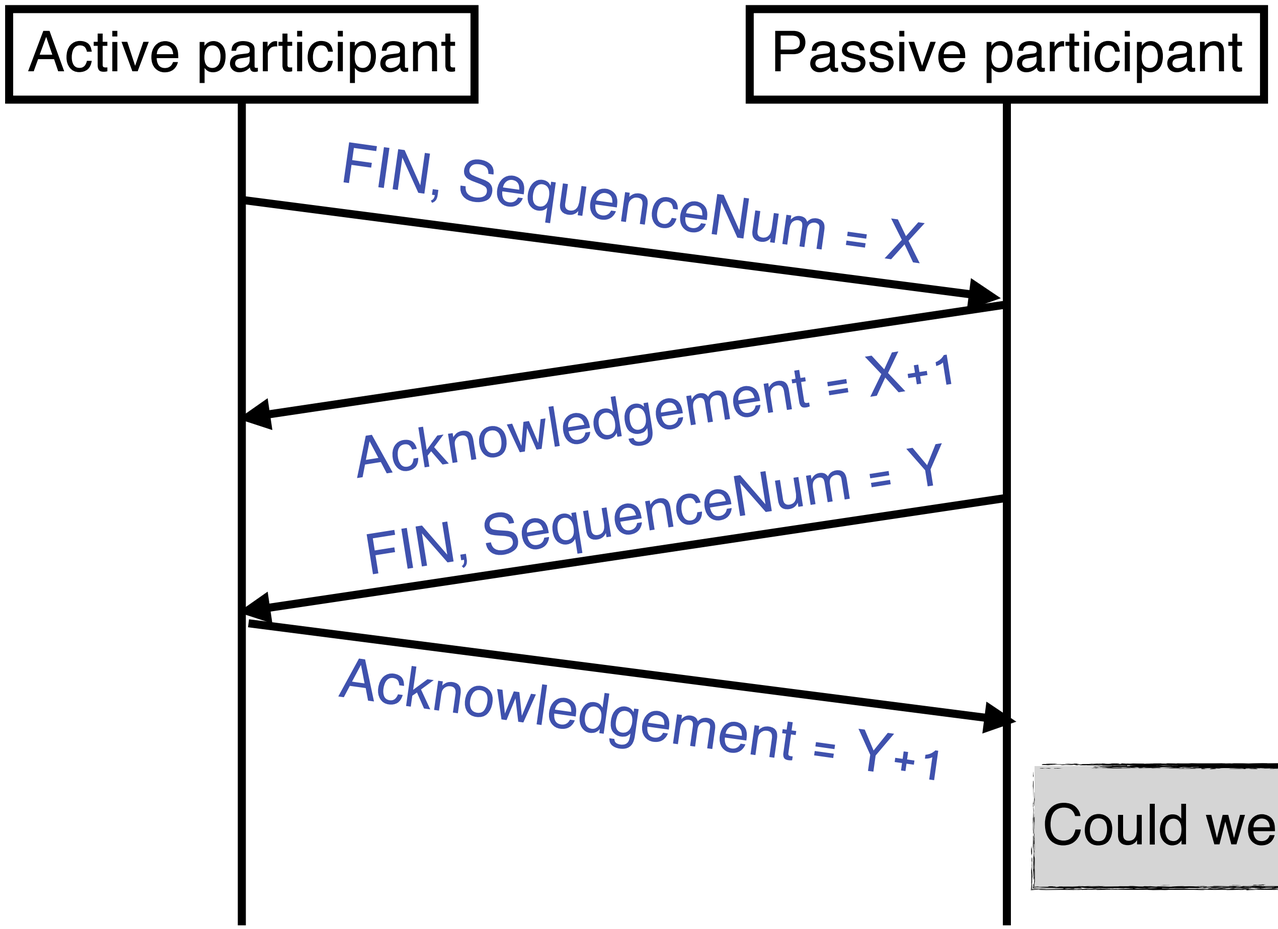
Case 1: One-side Closes First

4-way handshake



Case 1: One-side Closes First

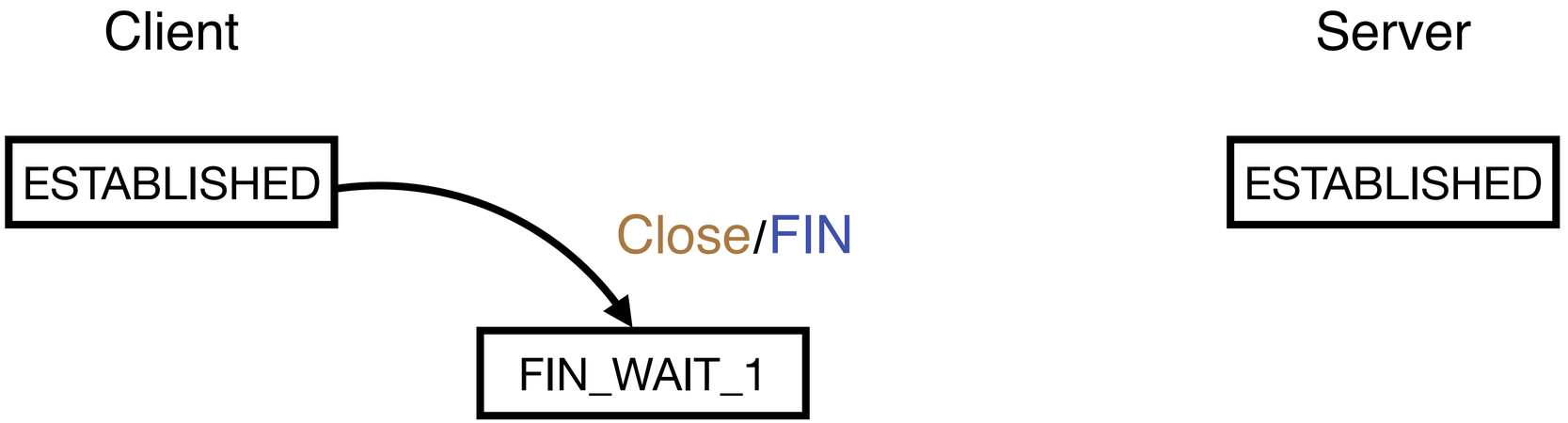
4-way handshake



Could we do a 3-way handshake?

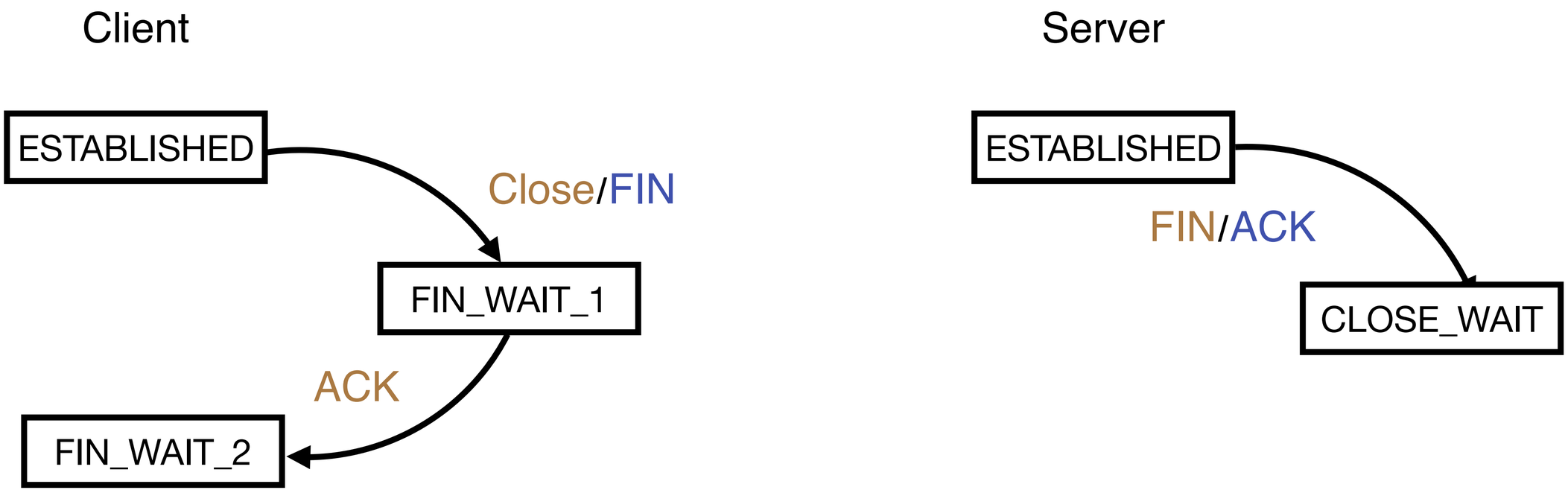
Case 1: State Machine Transition

Case 1: State Machine Transition (Step 1)

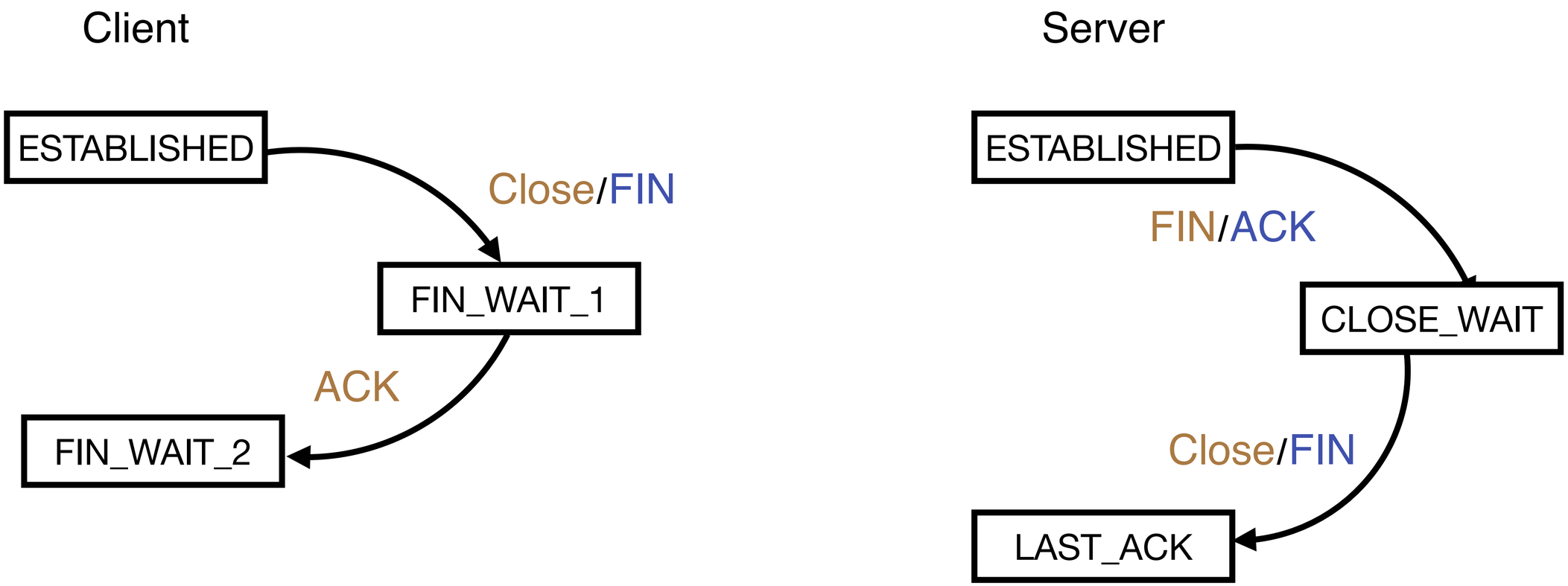


Case 1: State Machine Transition (Step 1)

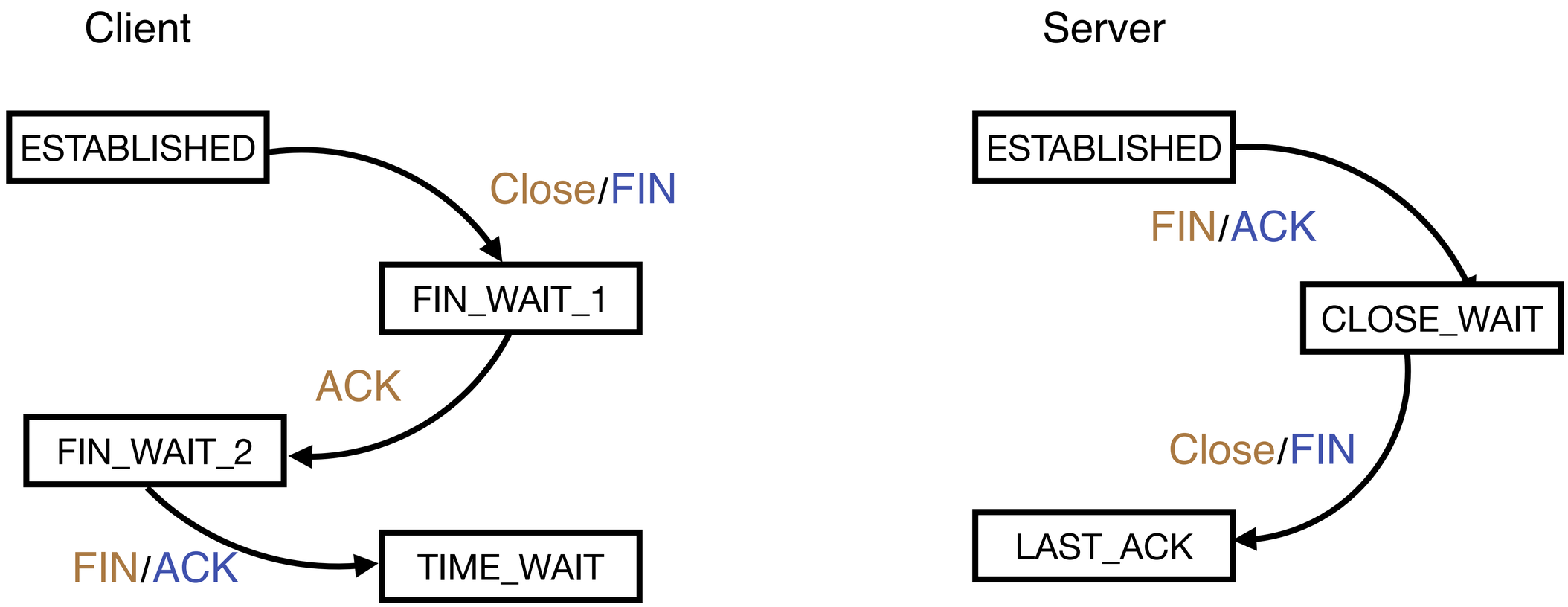
Case 1: State Machine Transition (Step 2)



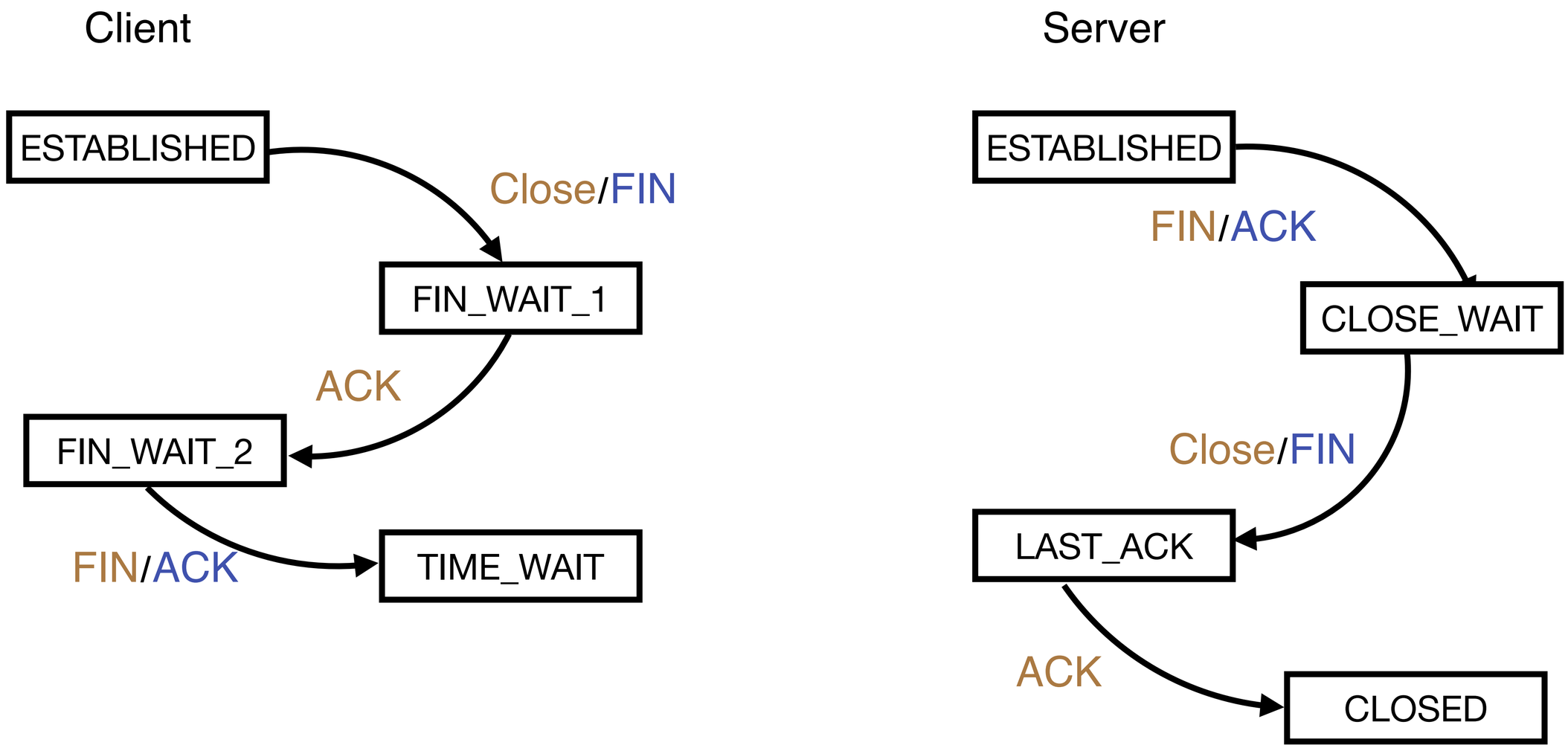
Case 1: State Machine Transition (Step 3)



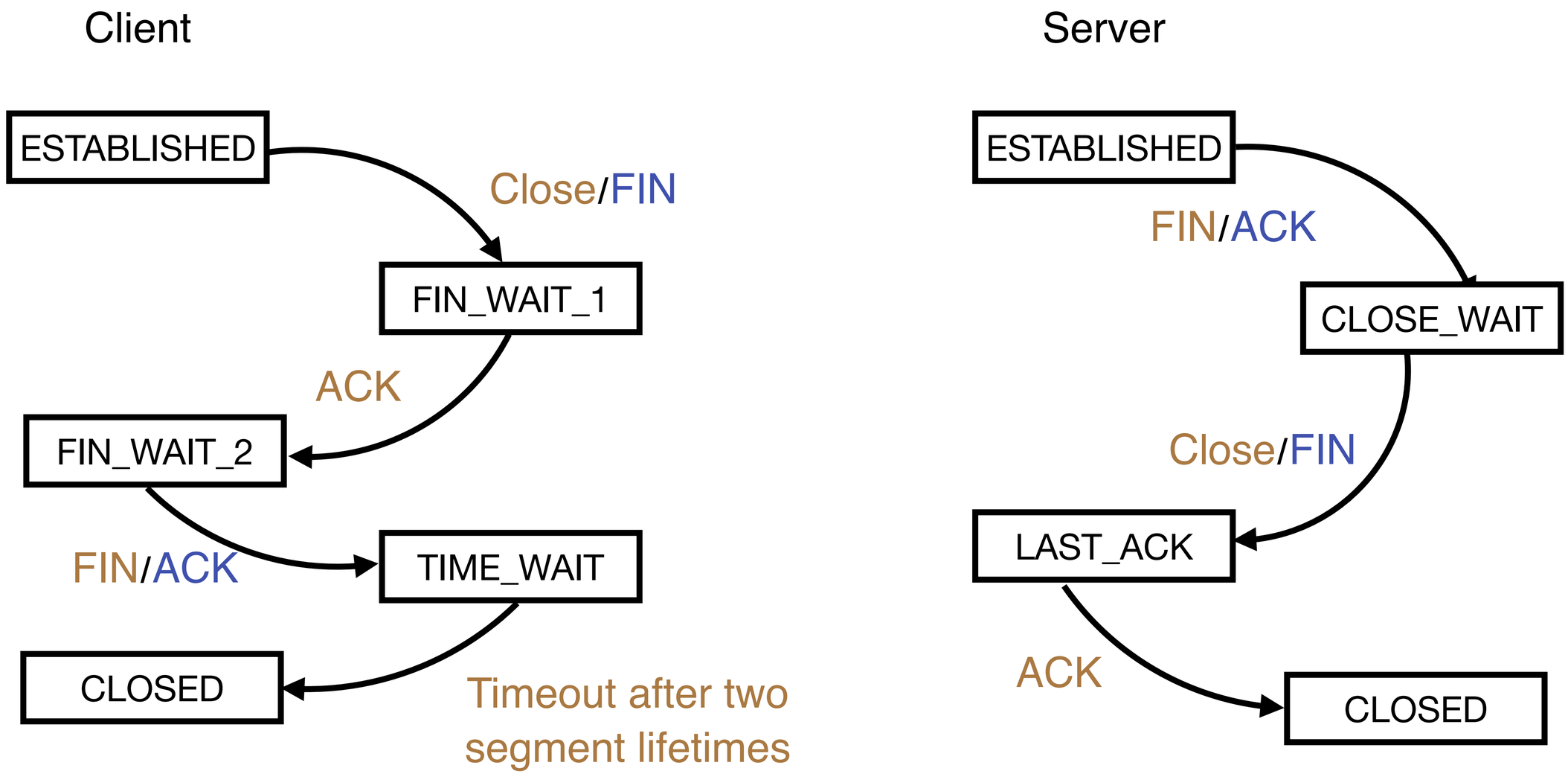
Case 1: State Machine Transition (Step 3)



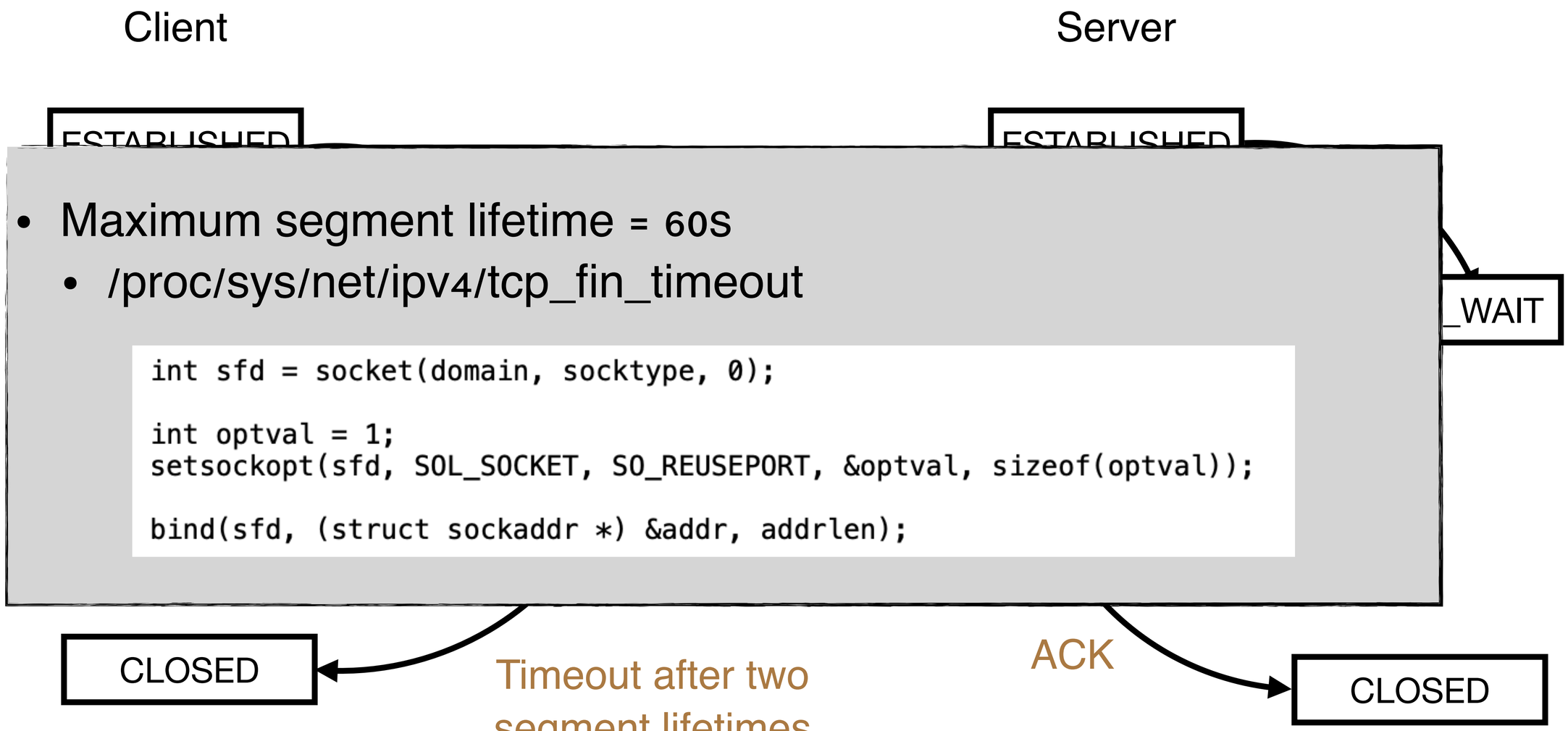
Case 1: State Machine Transition (Step 4)



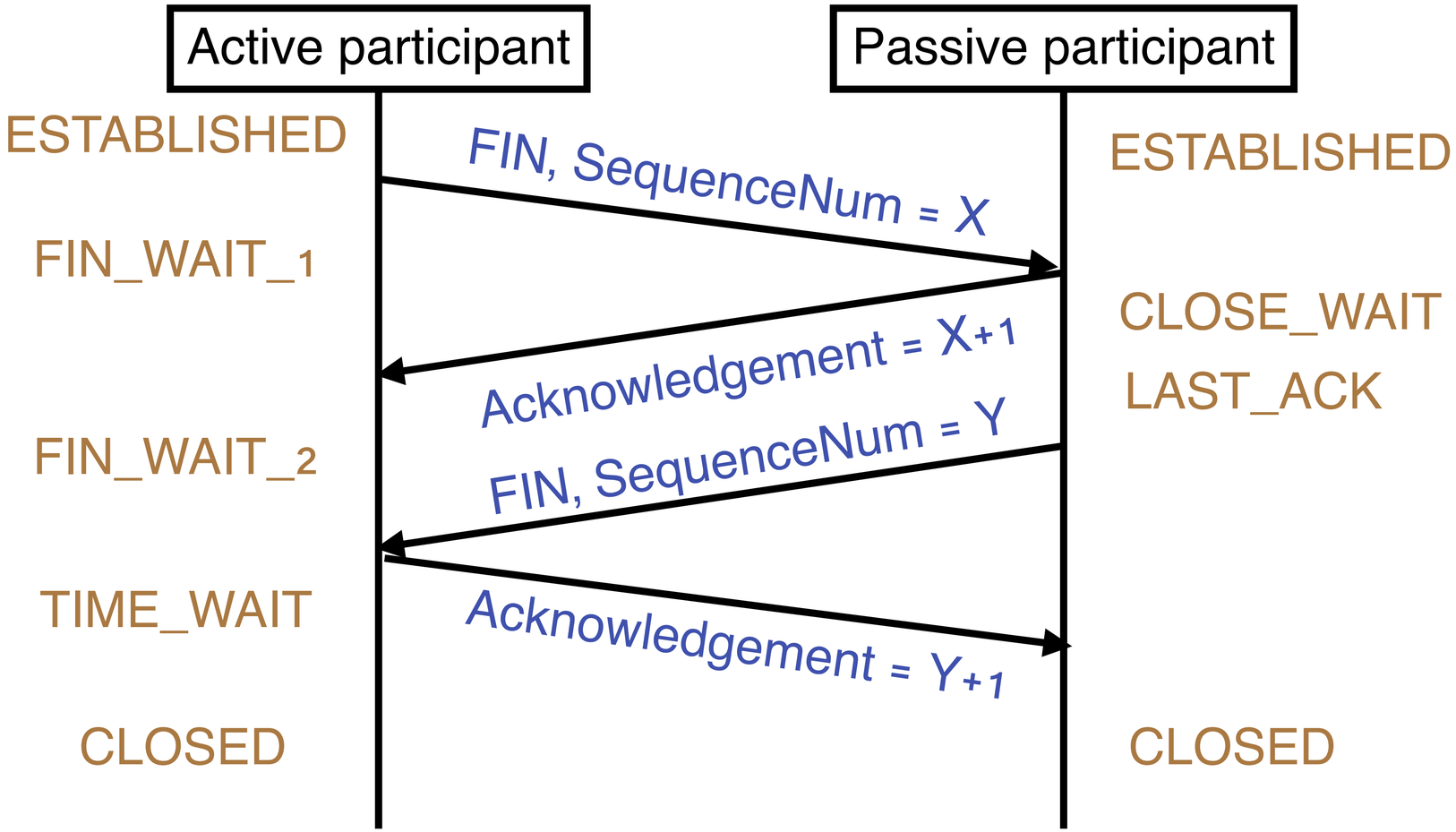
Case 1: State Machine Transition (Step 4)



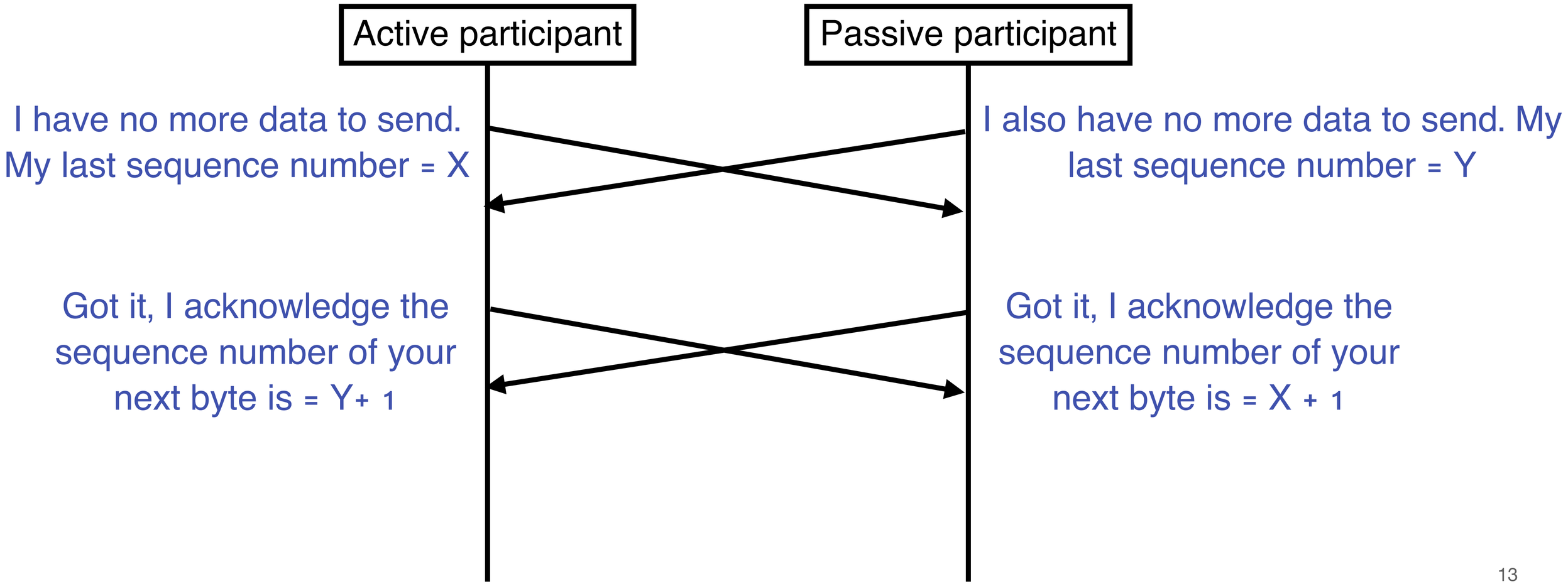
Case 1: State Machine Transition (Step 4)



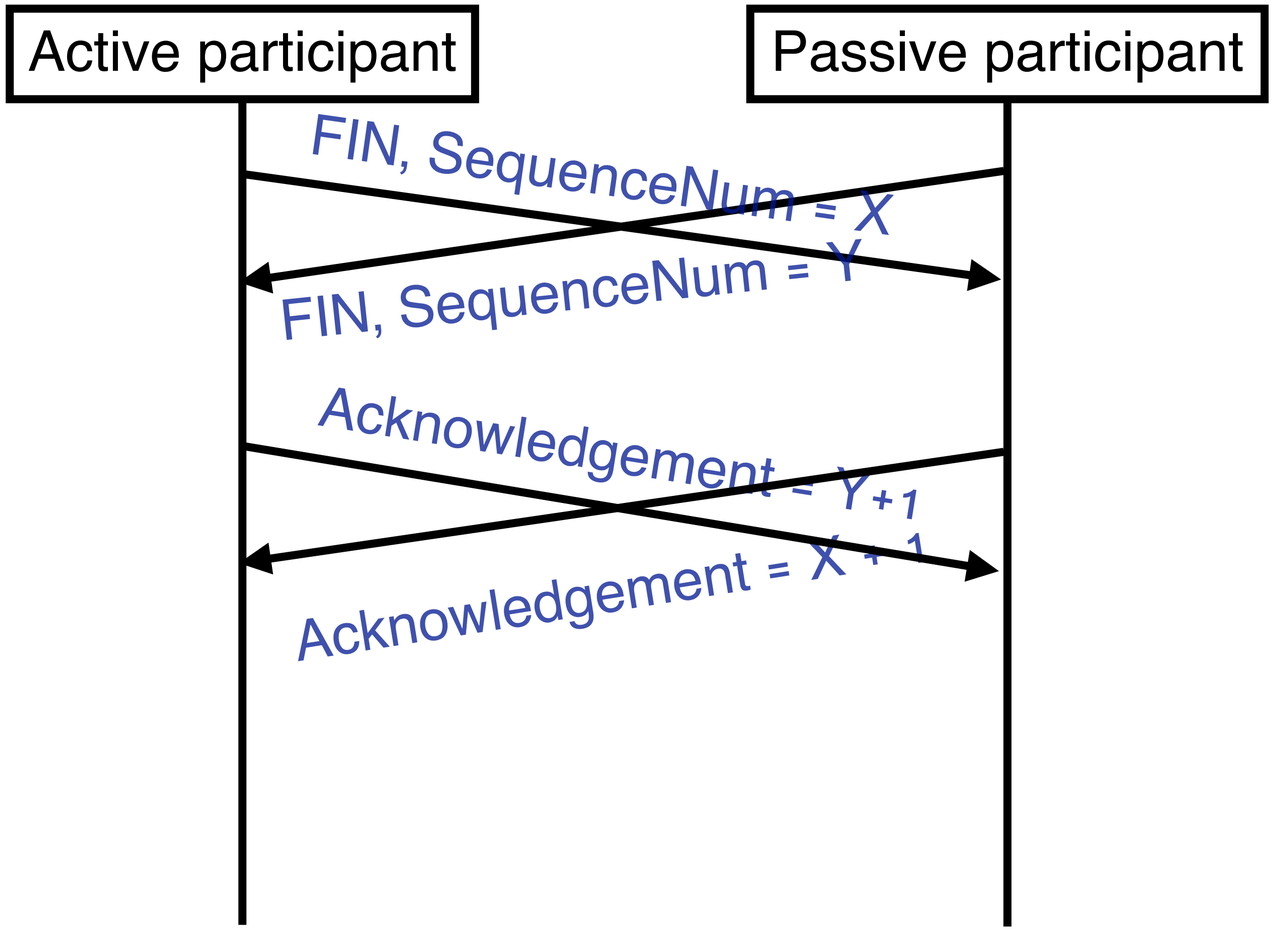
TCP Connection Termination (Case1) Summary



Case 2: Both Sides Close Simultaneously



Case 2: Both Sides Close Simultaneously



Case 2: State Machine Transition (Step 1)

Client

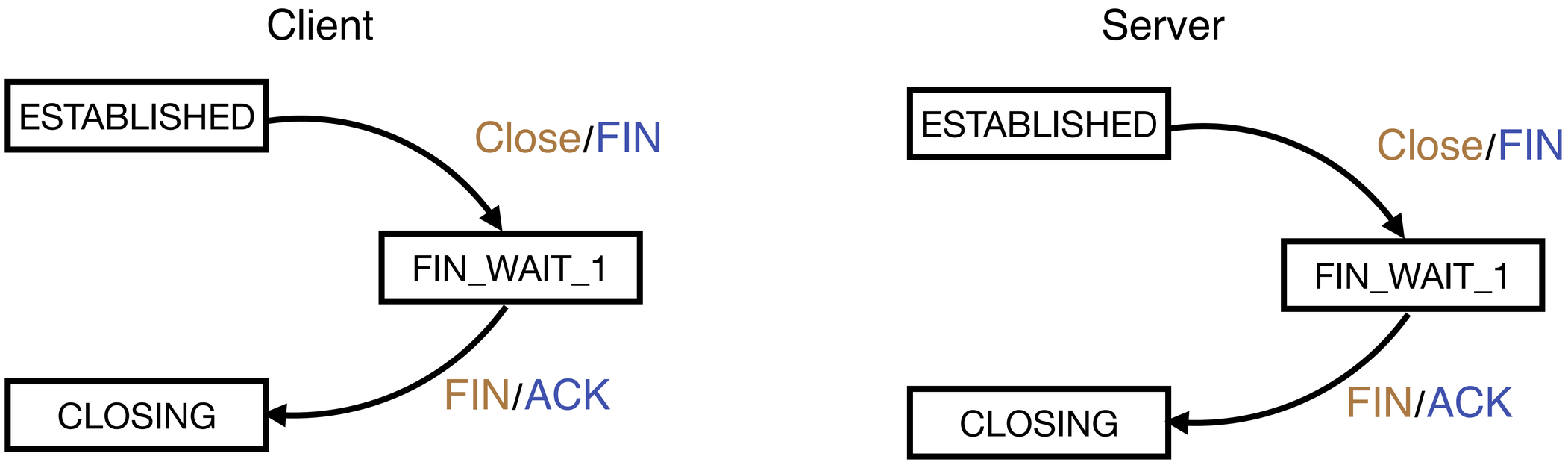
ESTABLISHED

Server

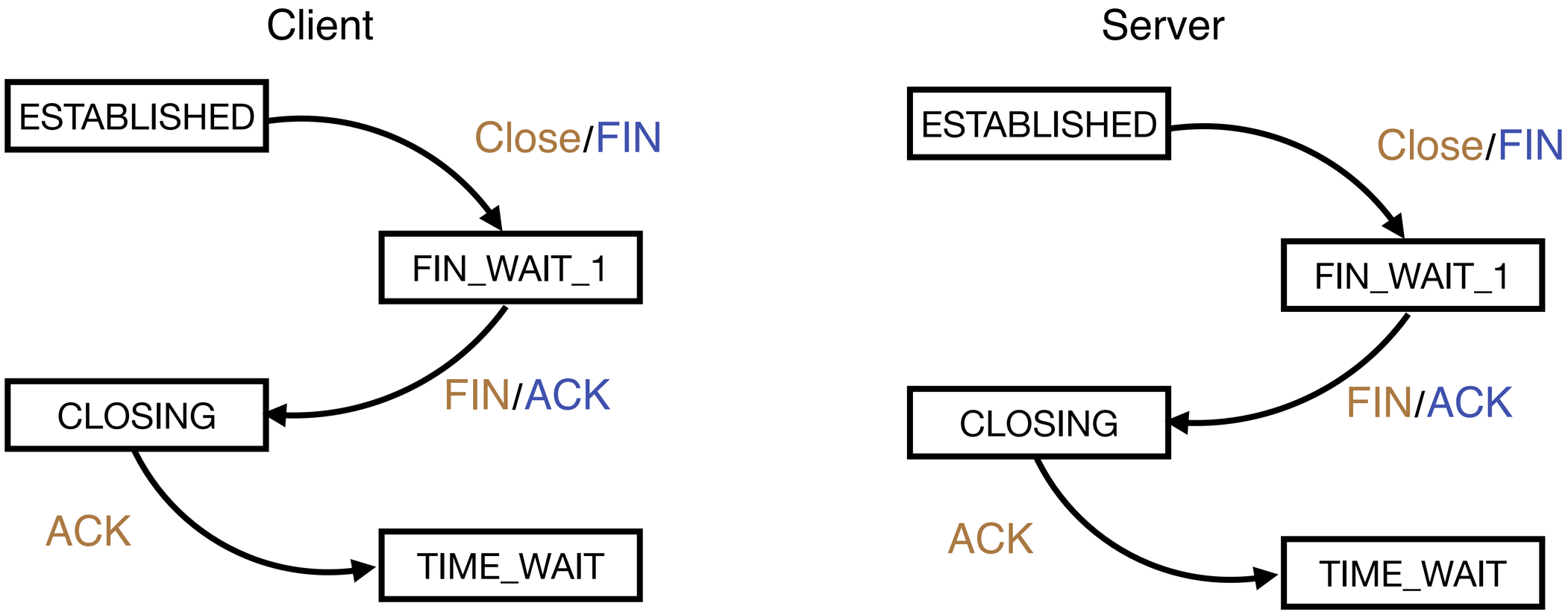
ESTABLISHED

Case 2: State Machine Transition (Step 1)

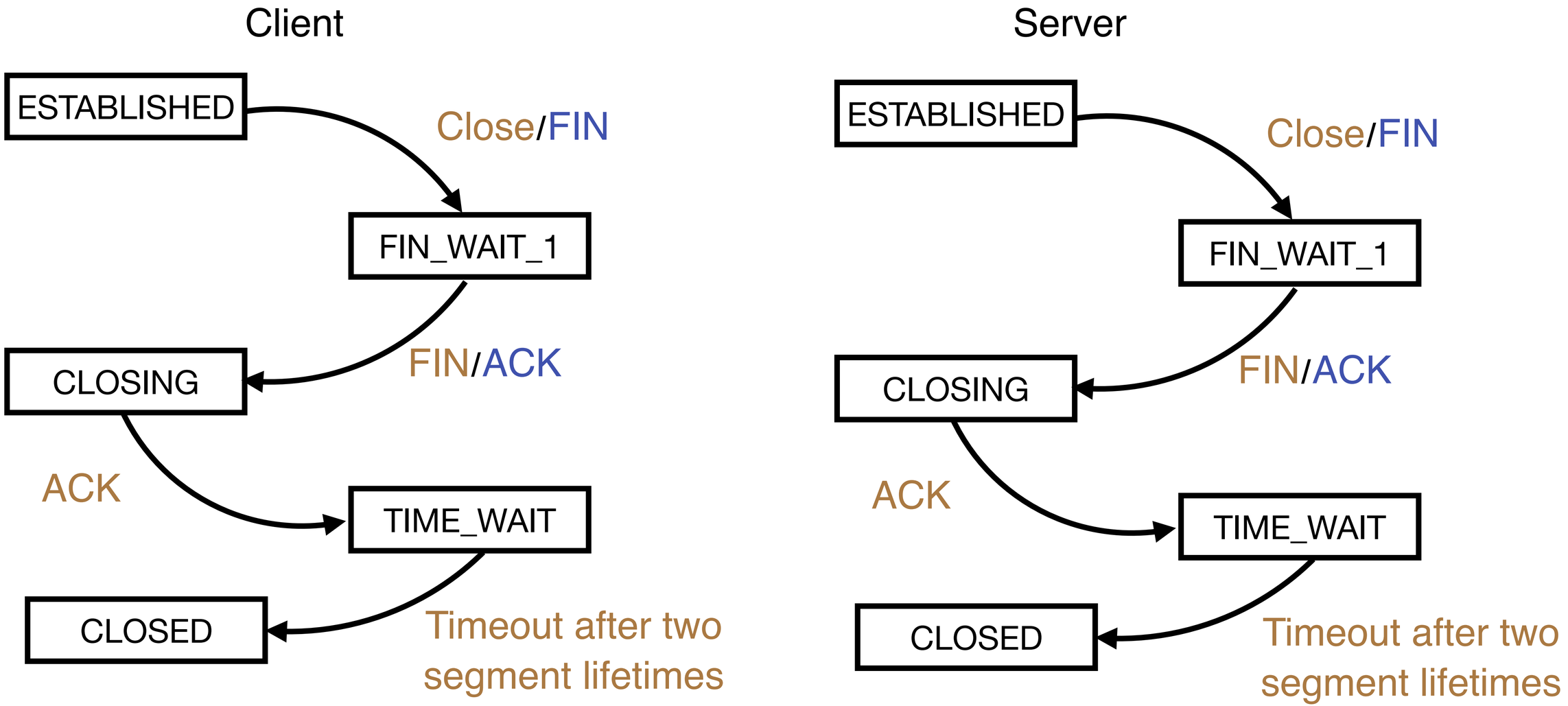
Case 2: State Machine Transition (Step 2)



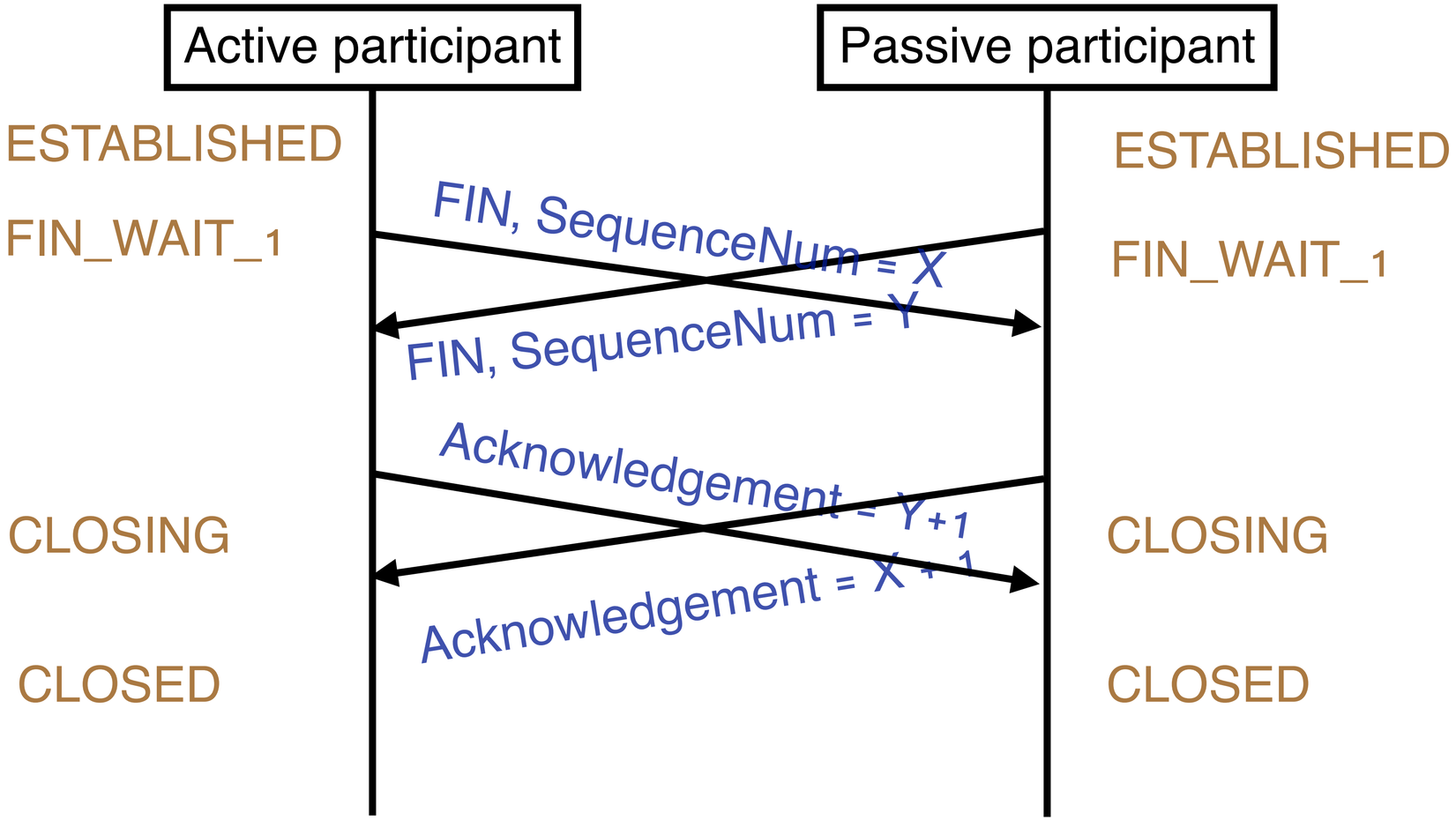
Case 2: State Machine Transition (Step 3)



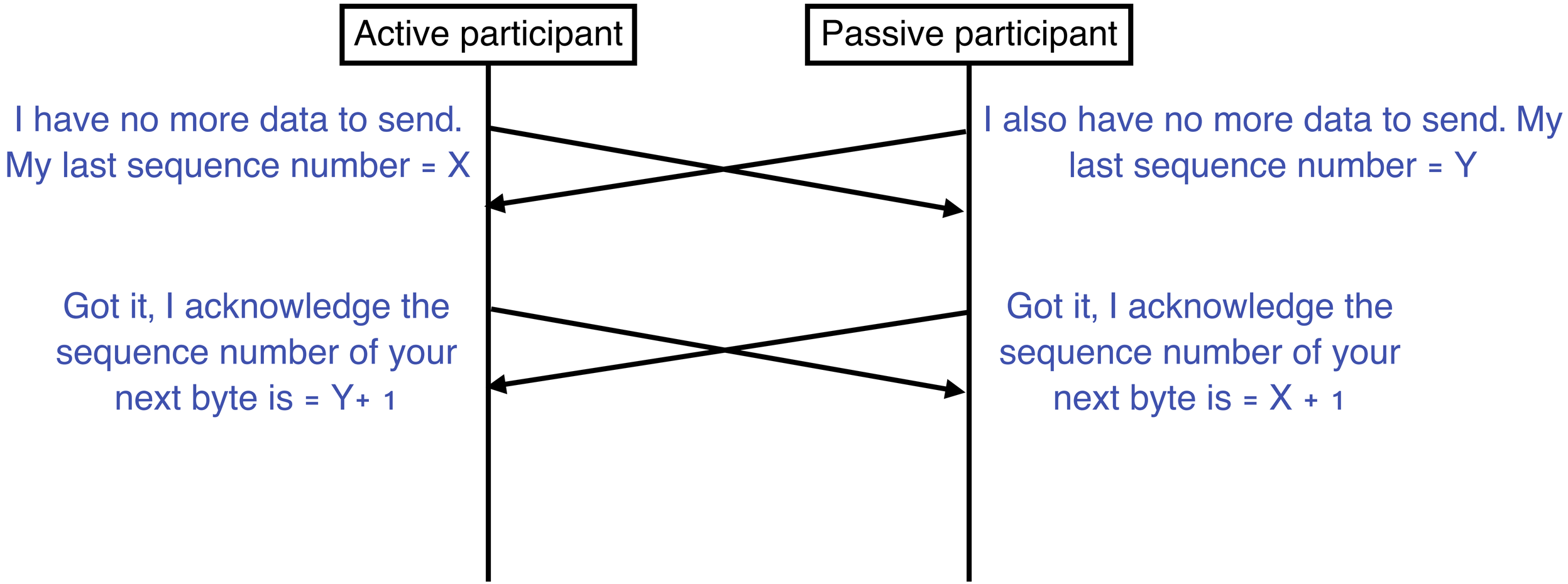
Case 2: State Machine Transition (Step 4)



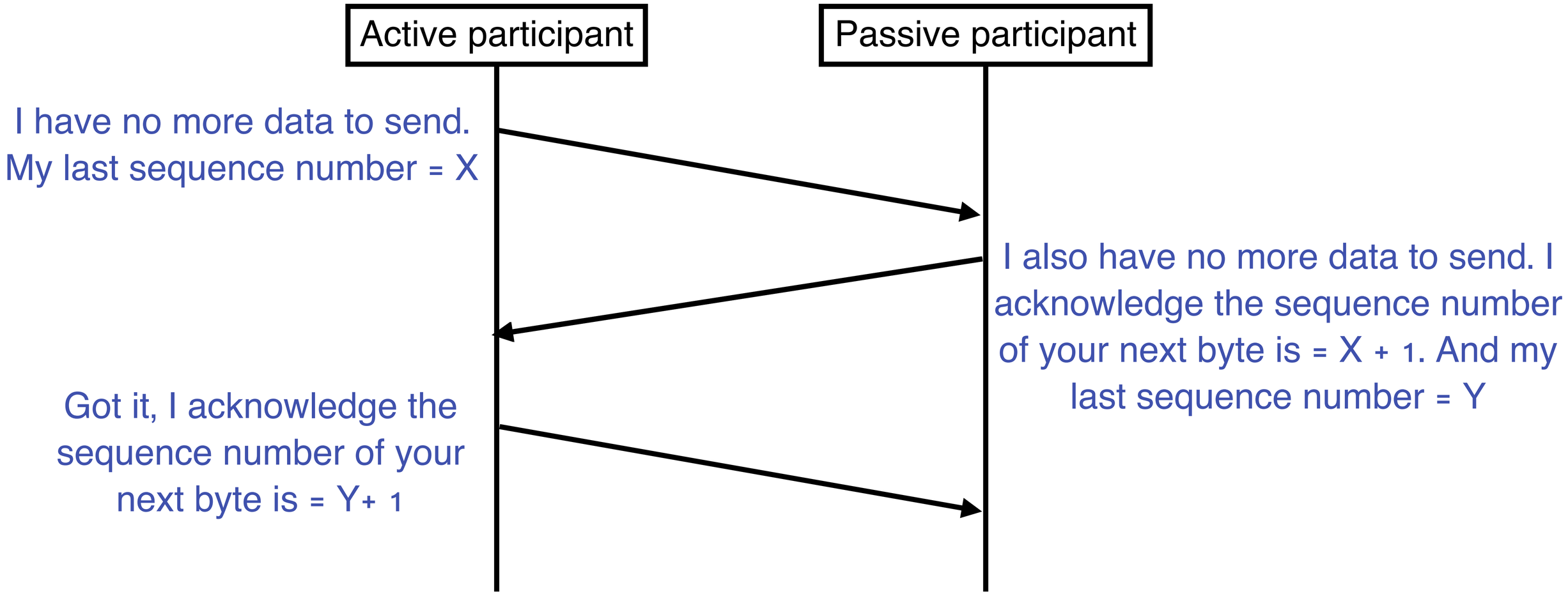
TCP Connection Termination (Case 2) Summary



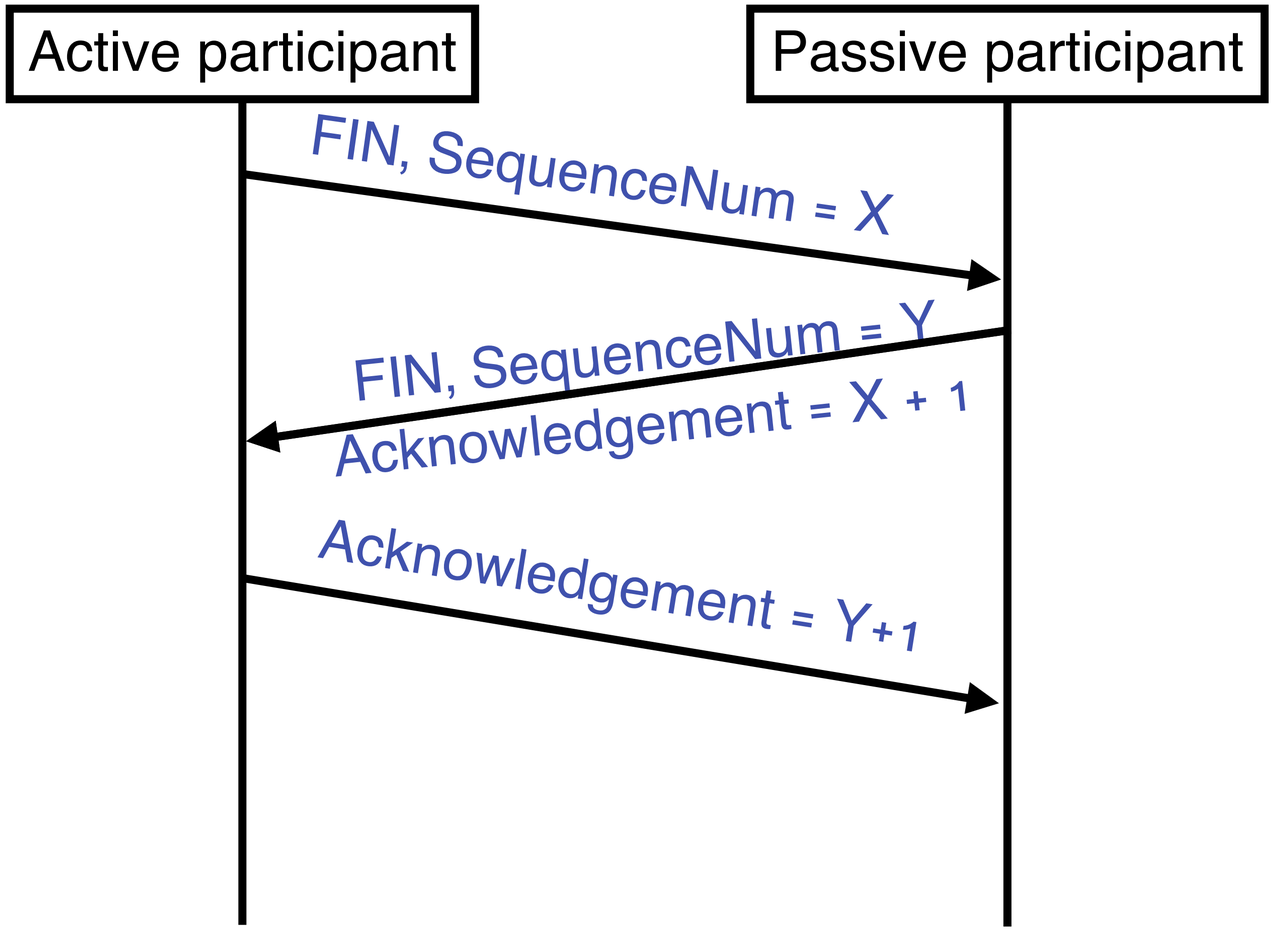
Case 3: Both Sides Close Simultaneously, but



Case 3: Both Sides Close Simultaneously, but



Case 3: Both Sides Close Simultaneously, but

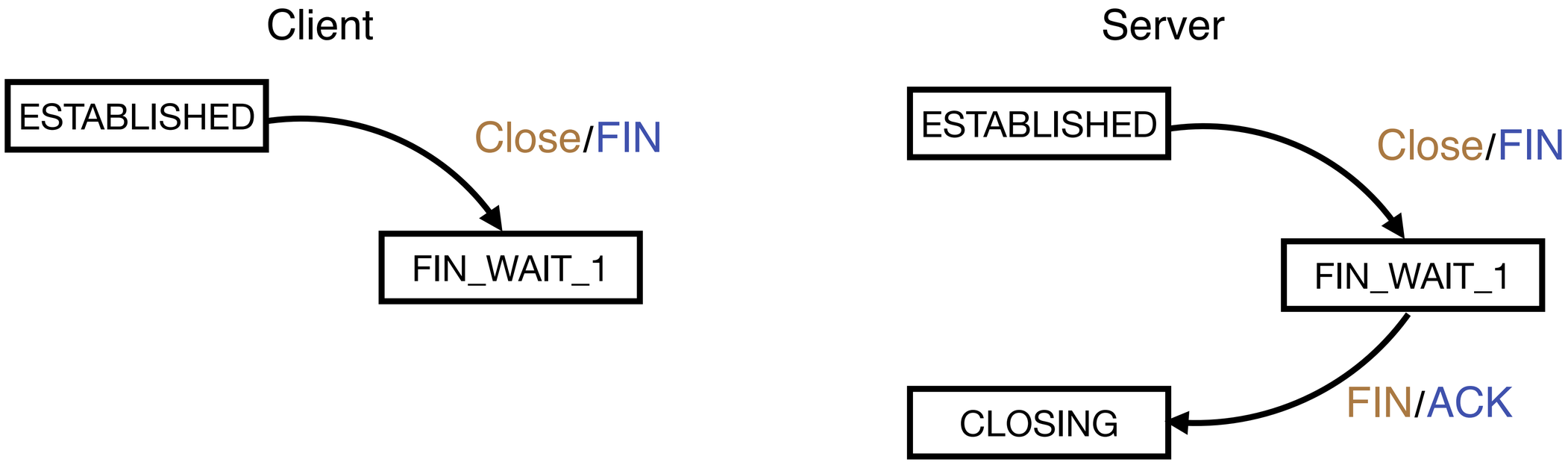


Case 3: State Machine Transition

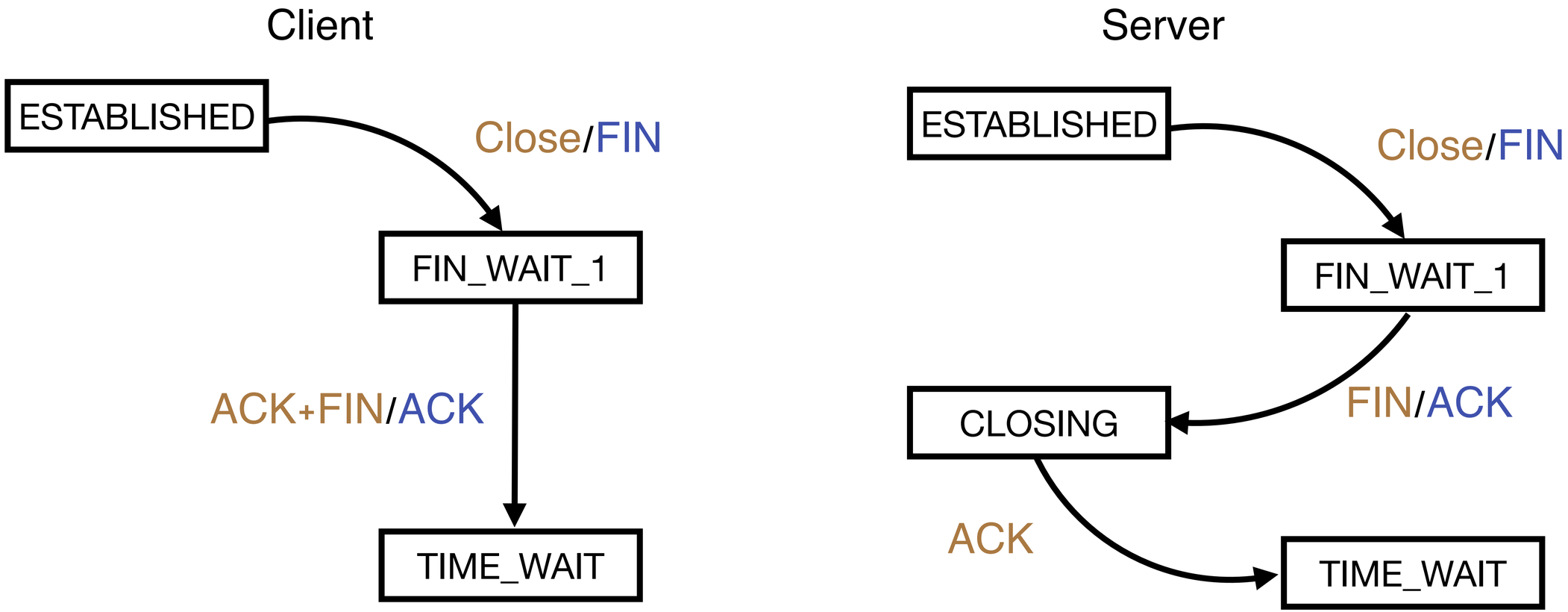
Client

Server

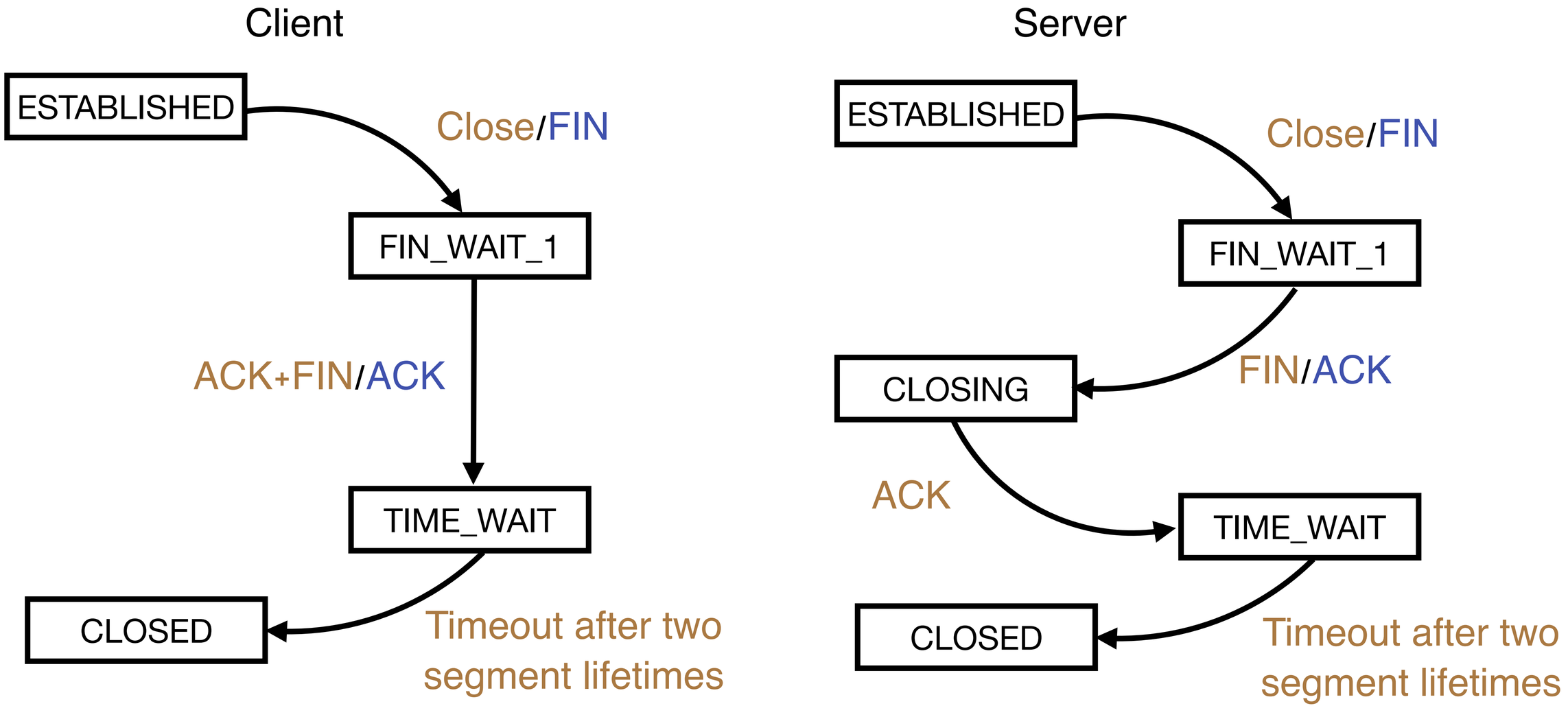
Case 3: State Machine Transition (Step 1)



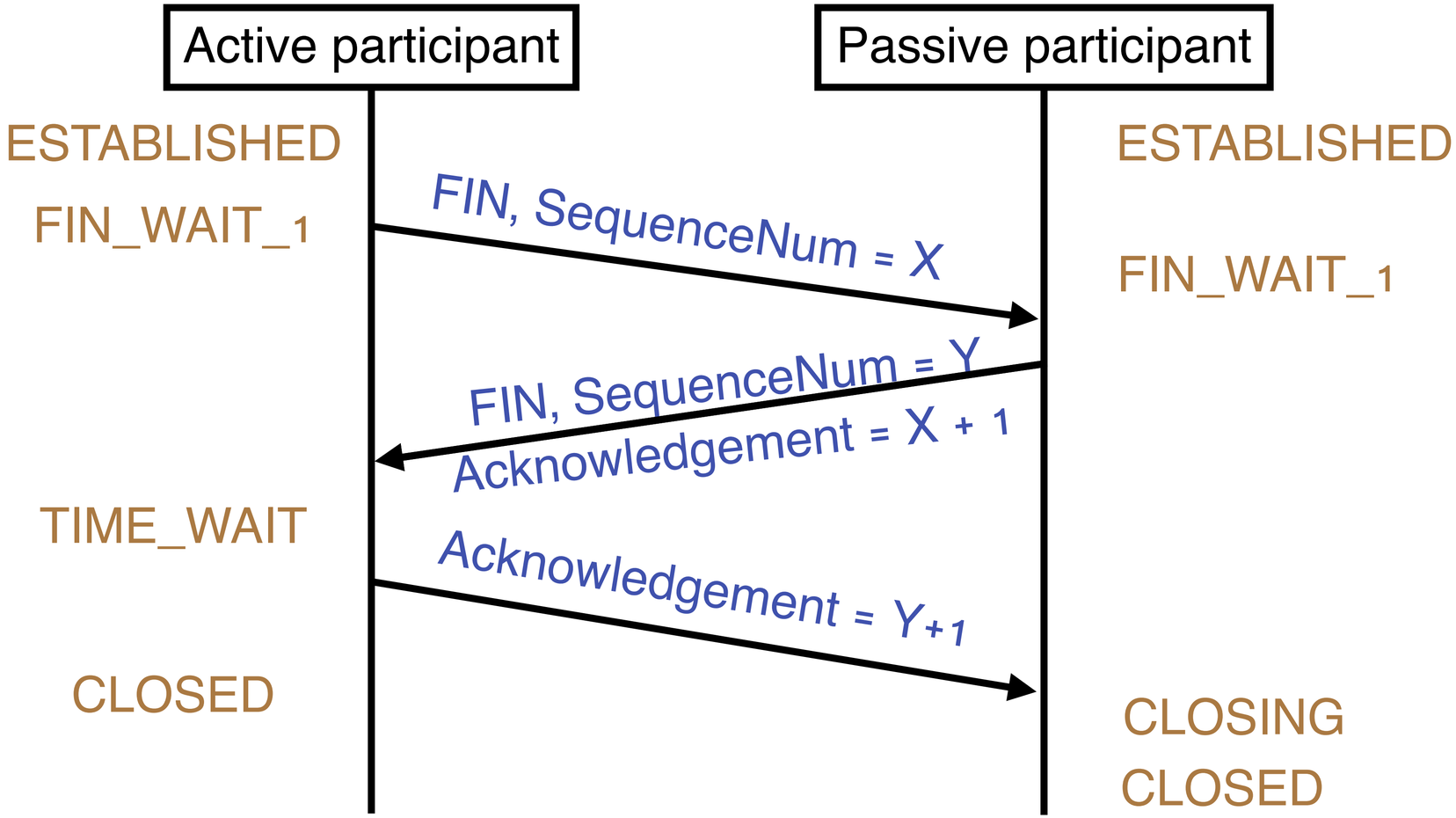
Case 3: State Machine Transition (Step 2)



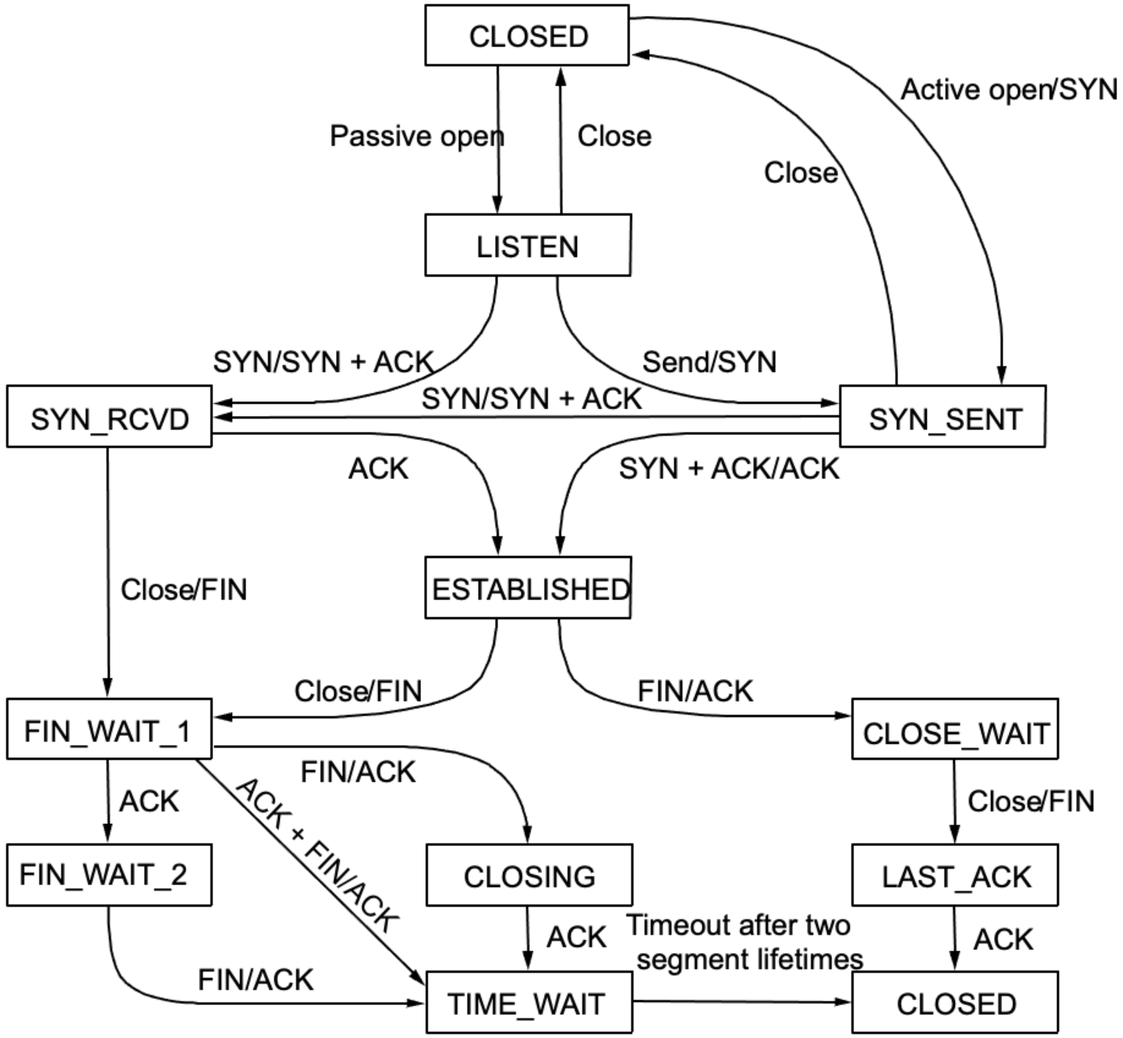
Case 3: State Machine Transition (Step 3)



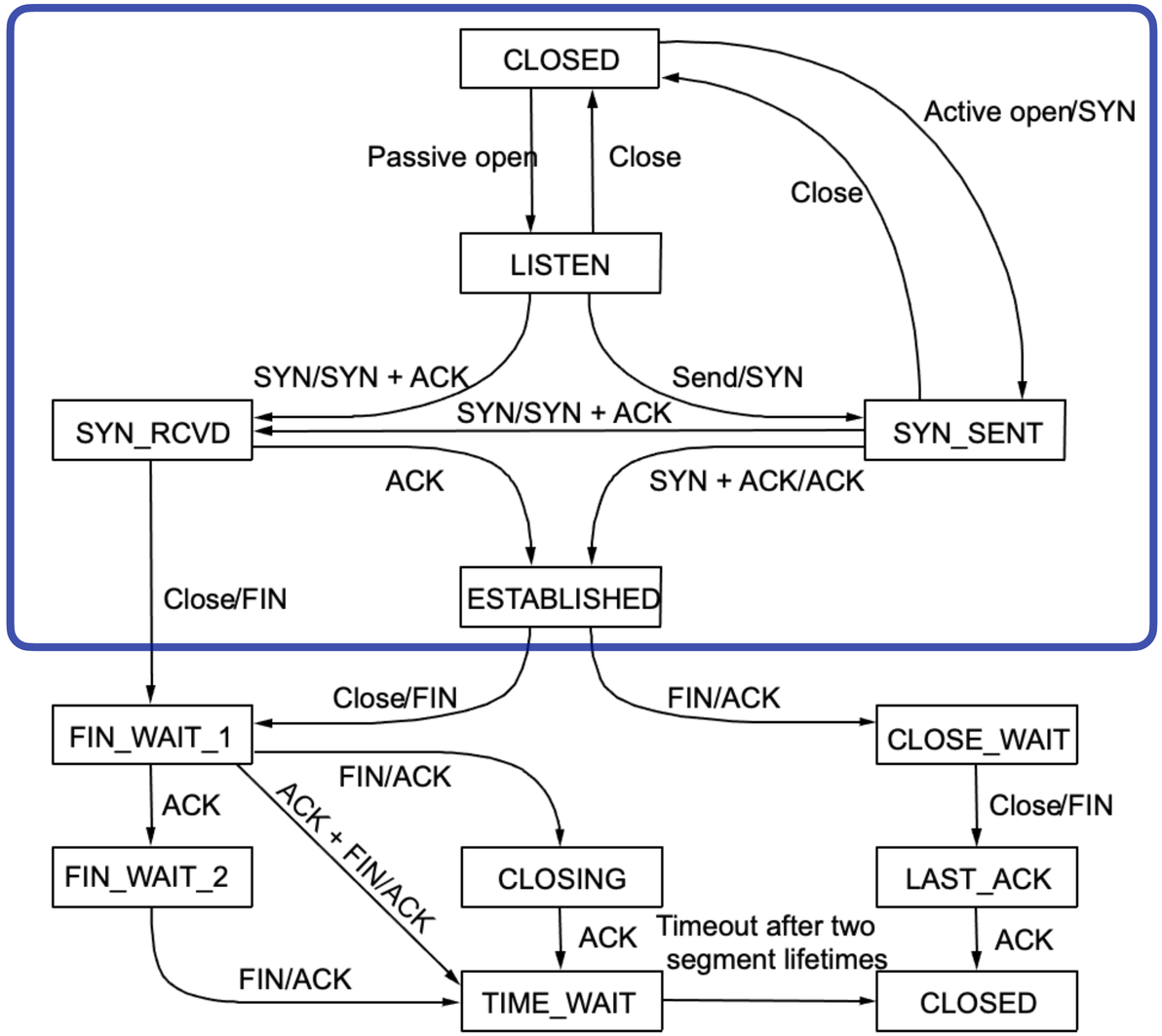
TCP Connection Termination (Case 3) Summary



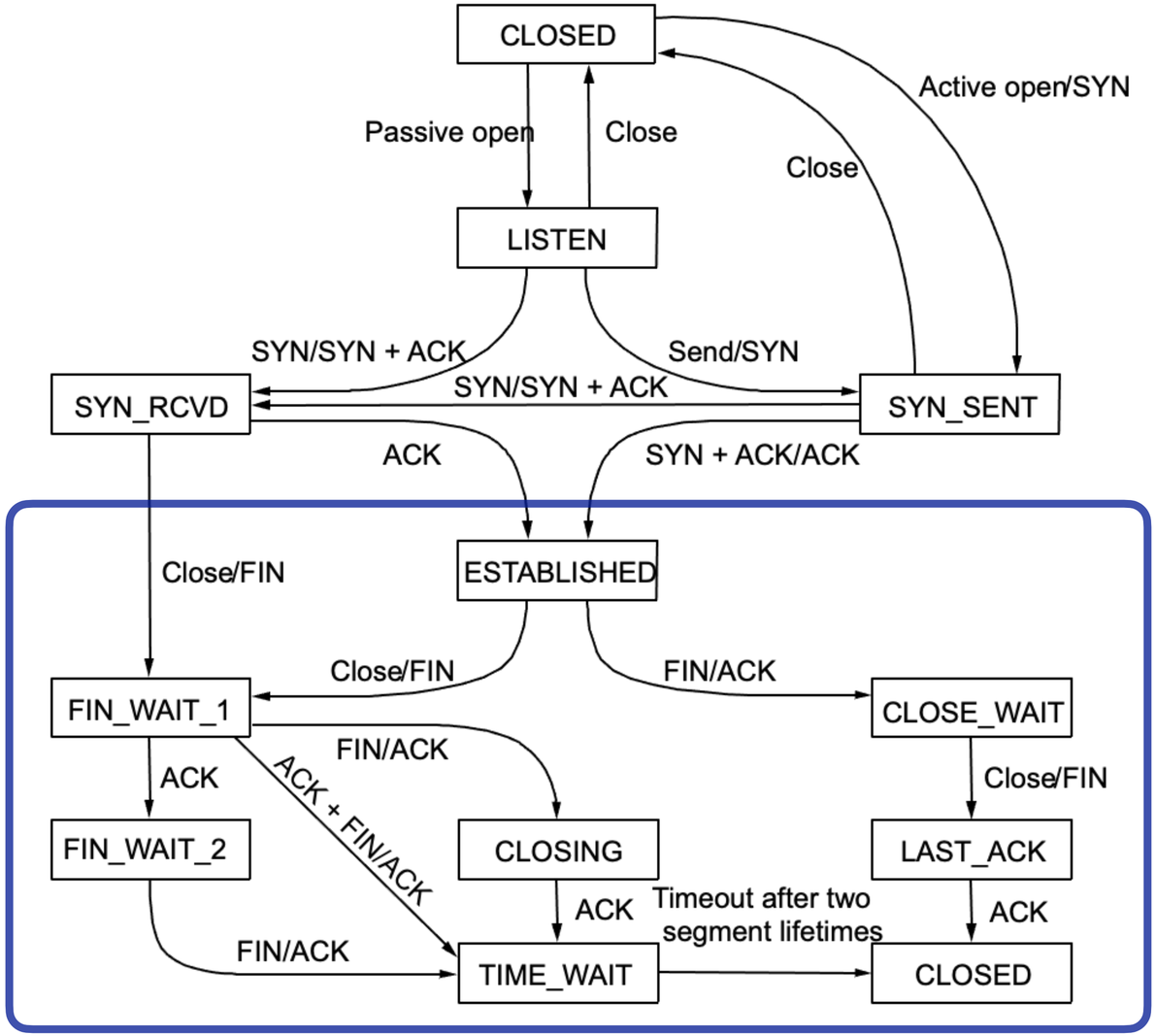
TCP State Transition Diagram Overall



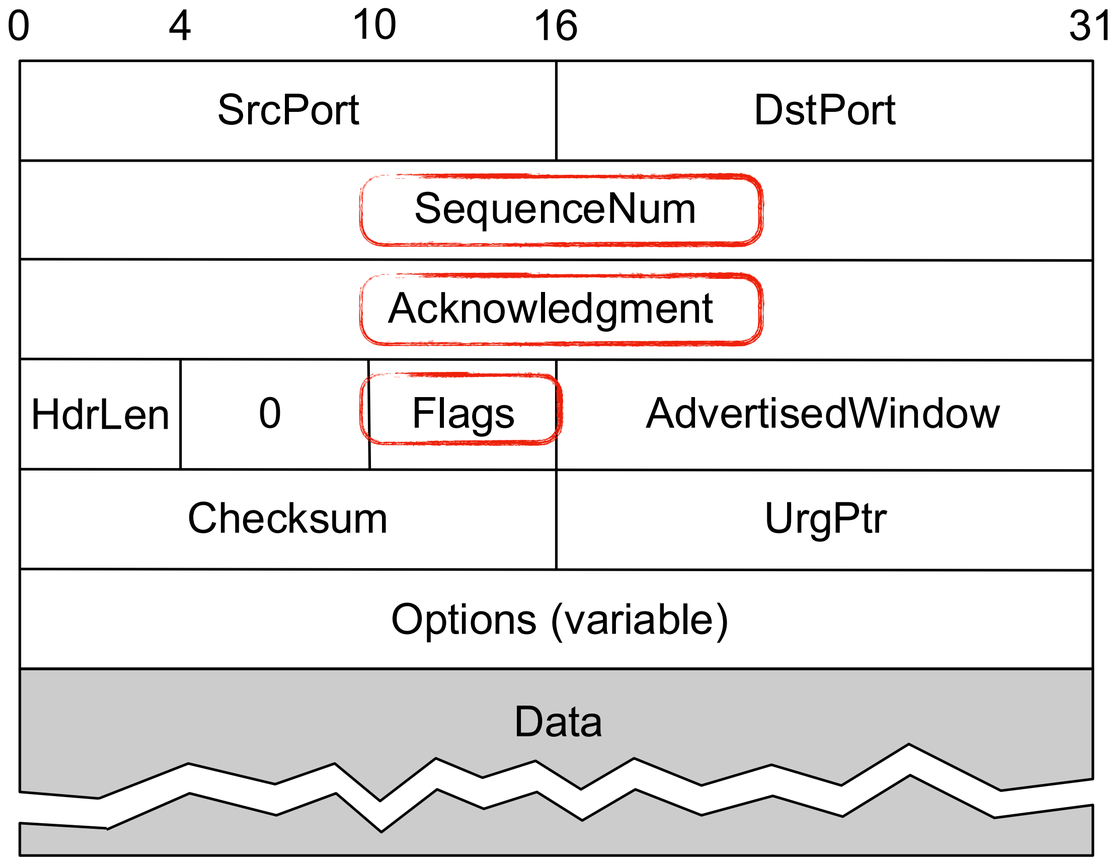
TCP State Transition Diagram Overall



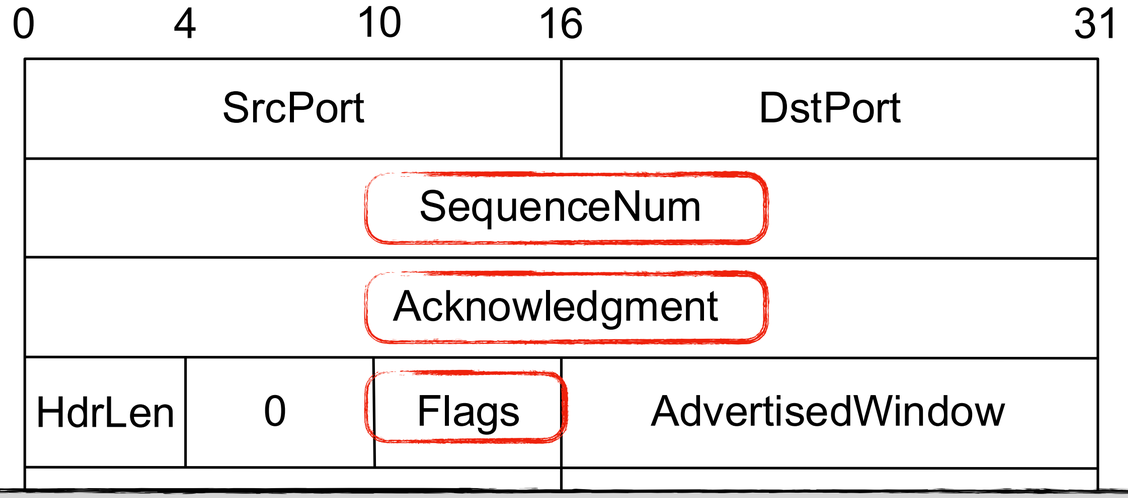
TCP State Transition Diagram Overall



Revisit the TCP Header



Revisit the TCP Header



- SYN/FIN –> TCP connection establishment and teardown
- ACK -> Acknowledgement is valid
- URG –> The segment contains urgent data. UrgPtr will be setup
- PUSH -> Notify the receiving process
- RESET -> The receiving side gets confused information

TCP Connection Management Summary

#1: Connection setup is asymmetric

- One side does a passive open the other side does an active open

TCP Connection Management Summary

#1: Connection setup is asymmetric

- One side does a passive open the other side does an active open

#2: Connection teardown is symmetric

- Each side has to close the connection independently

TCP Connection Management Summary

#1: Connection setup is asymmetric

- One side does a passive open the other side does an active open

#2: Connection teardown is symmetric

- Each side has to close the connection independently

#3: Most of the states schedule a timeout

- The timeout event is triggered when the expected response does not happen

TCP Connection Management Summary

#1: Connection setup is asymmetric

- One side does a passive open the other side does an active open

#2: Connection teardown is symmetric

- Each side has to close the connection independently

TCP(UDP) Connection = Flow

- The network processing granularity in the transport layer
- Five tuples = (src IP, dst IP, protocol number, src port, dst port)

How TCP solves the first issue?

#1: Arbitrary communication

- Senders and receivers can talk to each other in any ways

#2: No reliability guarantee

- Packets can be lost/duplicated/reordered during transmission
- Checksum is not enough

#3: No resource management

- Each communication channel works as an exclusive network resource owner
- No adaptiveness support for the physical networks and applications

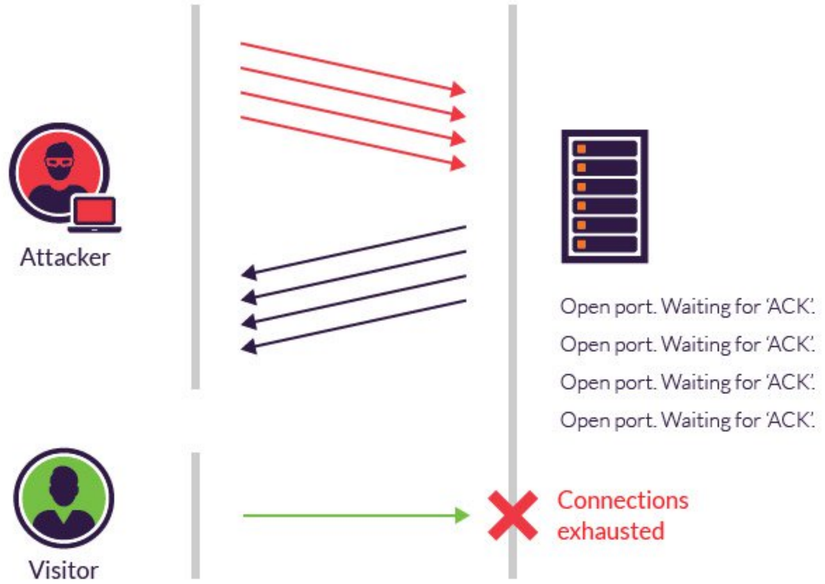
TCP avoids arbitrary communication but exposes non-negligible attacking interfaces.

SYN Flood

The TCP connection establishment phase starts with a standardized three-way handshake. The client sends an SYN packet. The server responds with an SYN-ACK.

SYN Flood

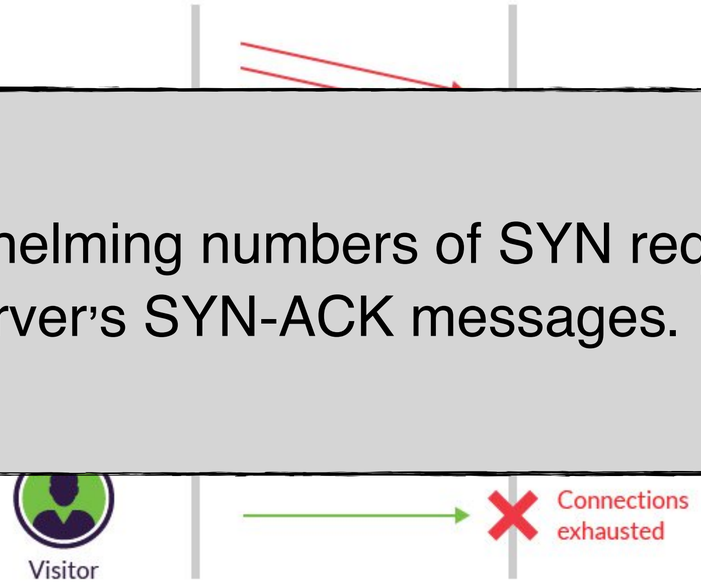
The TCP connection establishment phase starts with a standardized three-way handshake. The client sends an SYN packet. The server responds with an SYN-ACK.



SYN Flood

The TCP connection establishment phase starts with a standardized three-way handshake. The client sends an SYN packet. The server responds with an SYN-ACK.

An attacker sends overwhelming numbers of SYN requests and intentionally never responds to the server's SYN-ACK messages.



Terminology

1. Host
2. NIC
3. Multi-port I/O bridge
4. Protocol
5. RTT
6. Packet
7. Header
8. Payload
9. BDP
10. Baud rate
11. Frame/Framing
12. Parity bit
13. Checksum
14. Ethernet
15. MAC
16. (L2) Switch
17. Broadcast
18. Acknowledgement
19. Timeout
20. Datagram
21. TTL
22. MTU
23. Best effort
24. (L3) Router
25. Subnet mask
26. CIDR
27. Converge
28. Count-to-infinity
29. Line card
30. Network processor
31. Gateway
32. Private network
33. IPv6
34. Multicast
35. IGMP
36. SDN
37. (Transport) port
38. Pseudo header
39. SYN/ACK
40. Incarnation
41. Flow
42. SYN flood

Principle

1. Layering
2. Minimal States
3. Hierarchy

Technique

1. NRZ Encoding
2. NRZI Encoding
3. Manchester Encoding
4. 4B/5B Encoding
5. Byte Stuffing
6. Byte Counting
7. Bit Stuffing
8. 2-D Parity
9. CRC
10. MAC Learning
11. Store-and-Forward
12. Cut-through
13. Spanning Tree
14. CSMA/CD
15. Stop-and-Wait
16. Sliding Window
16. Fragmentation and Reassembly
17. Path MTU discovery
18. DHCP
19. Subnetting
20. Supernetting
21. Longest prefix match
22. Distance vector routing (RIP)
23. Link state routing (OSPF)
24. Boarder gateway protocol (BGP)
25. Network address translation (NAT)
26. User Datagram Protocol (UDP)
27. Transmission Control Protocol (TCP)
28. Three-way Handshake
29. TCP state transition

Summary

Today's takeaways

#1: TCP teardown is symmetric and presents three different cases

#2: TCP introduces a number of running states to deal with different kinds of communication scenarios

Next lecture

- TCP reliability support