Introduction to Computer Networks

TCP Connection Management (II)

https://pages.cs.wisc.edu/~mgliu/CS640/S25/index.html

Ming Liu
mgliu@cs.wisc.edu

Outline

- Last
  - TCP Connection Management (I)
- Today
  - TCP Connection Management (II)
- Announcements
  - Lab3 due on ~~04/01/2025~~ 04/03/2025 12:01PM
  - Quiz3 in class on 04/03/2025

Recap: UDP Issues

- **#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
  - A checksum is not enough

- #3: No resource management
  - Each channel works as an exclusive network resource owner
  - No adaptive support for the physical networks and applications

What is the goal of TCP connection management?

What is the goal of TCP connection management?

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

What is the goal of TCP connection management?

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

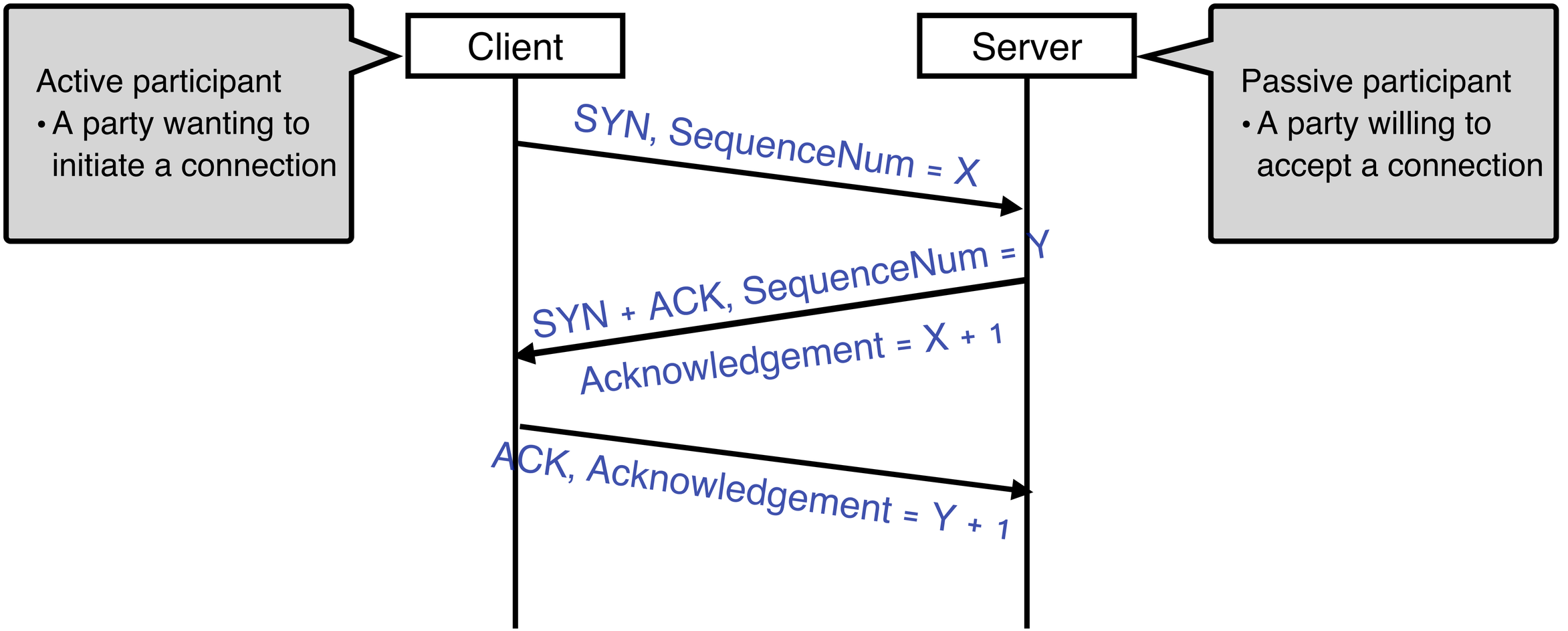
What is the goal of TCP connection management?

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

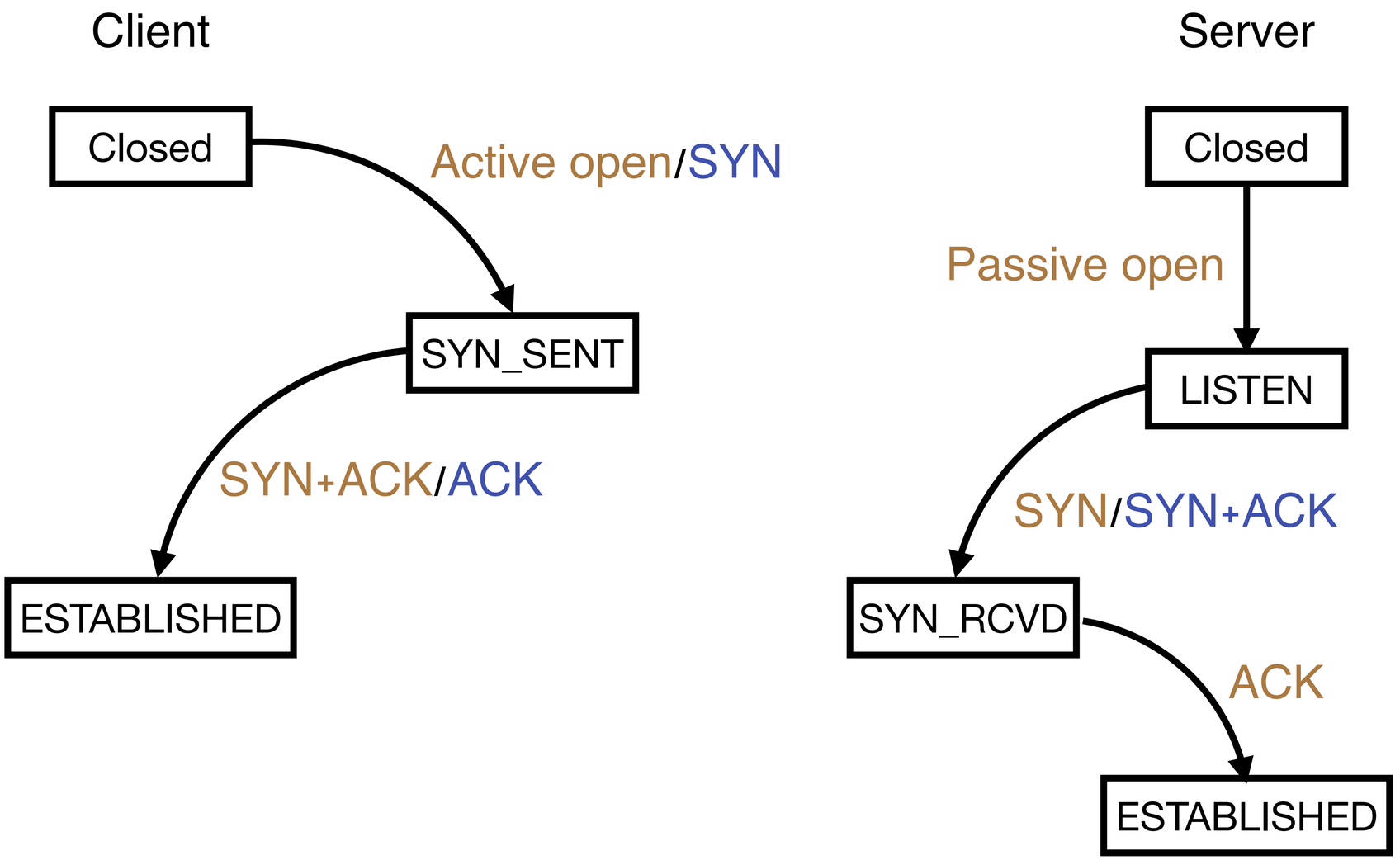
What is the goal of TCP connection management?

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

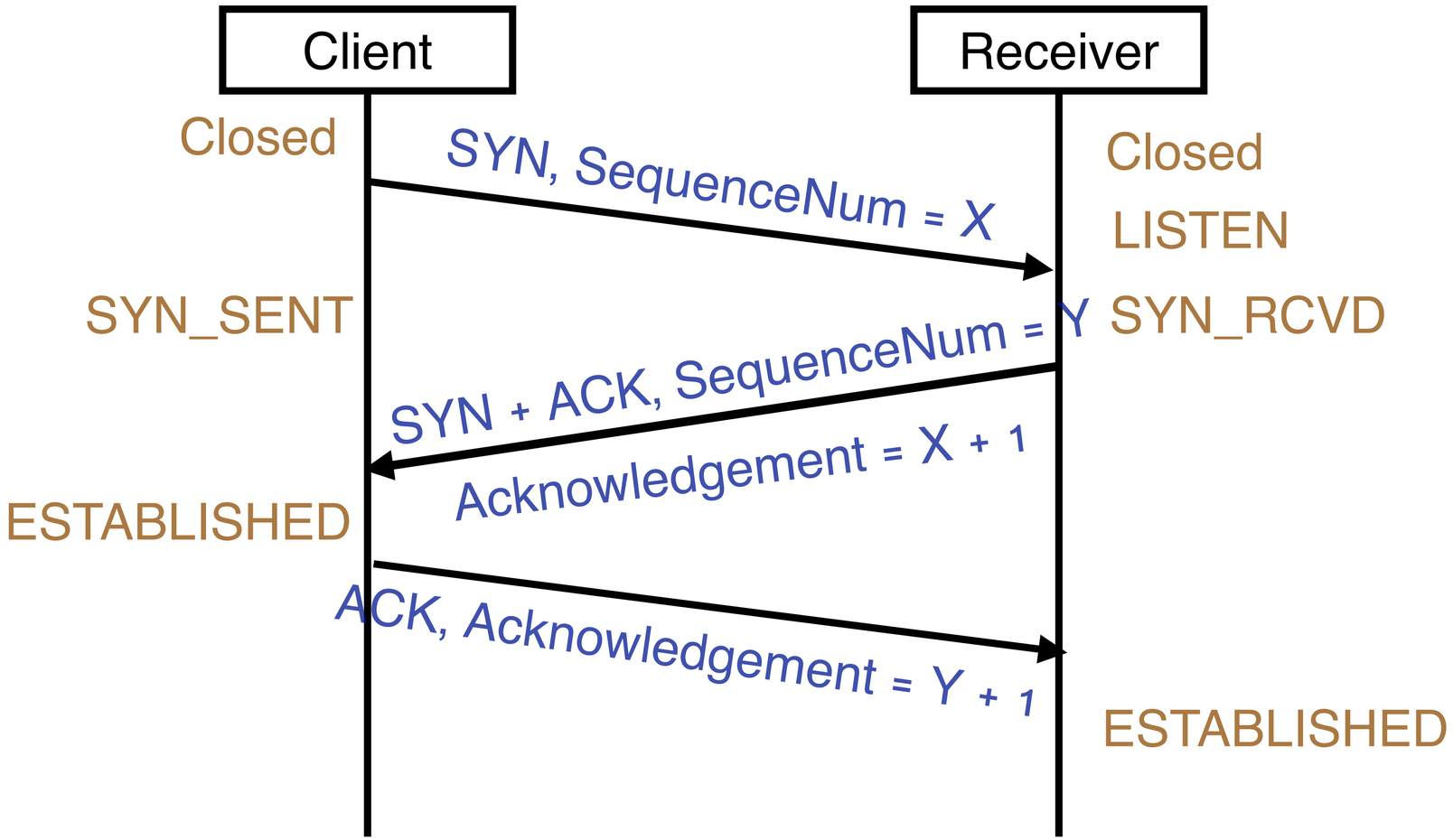
Three-Way Handshake



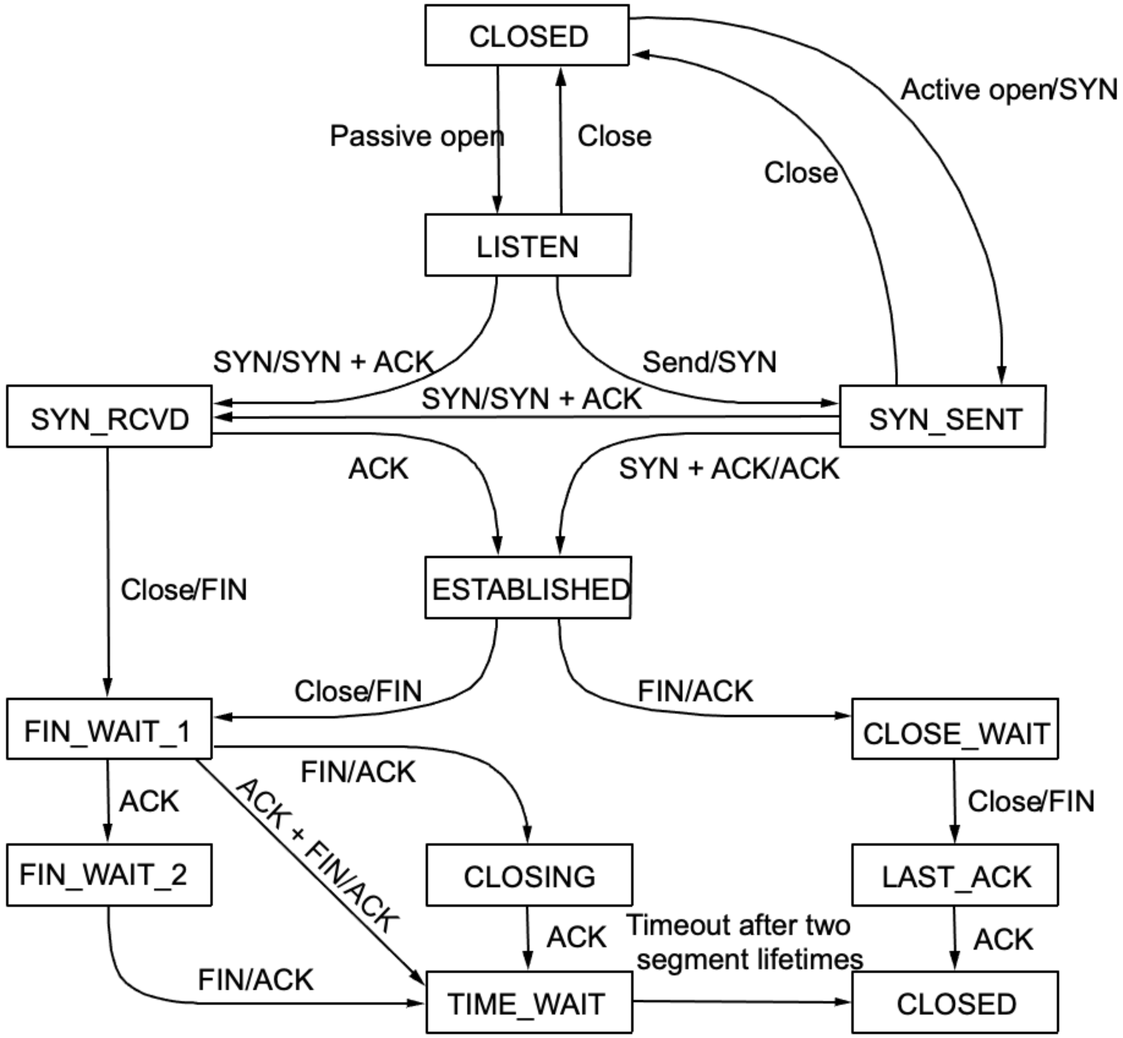
State Machine Transition



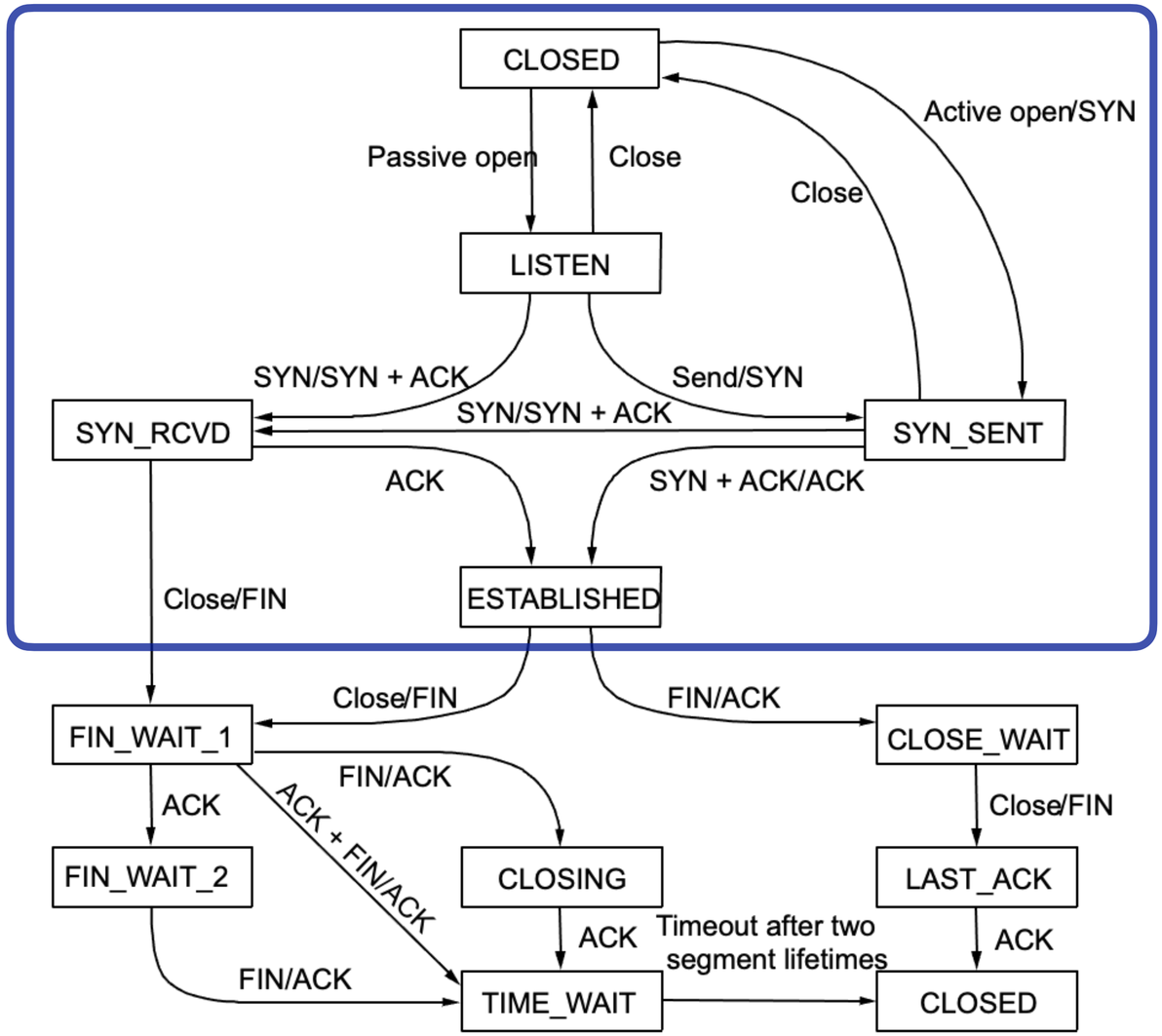
TCP Connection Establishment



TCP State Transition Diagram

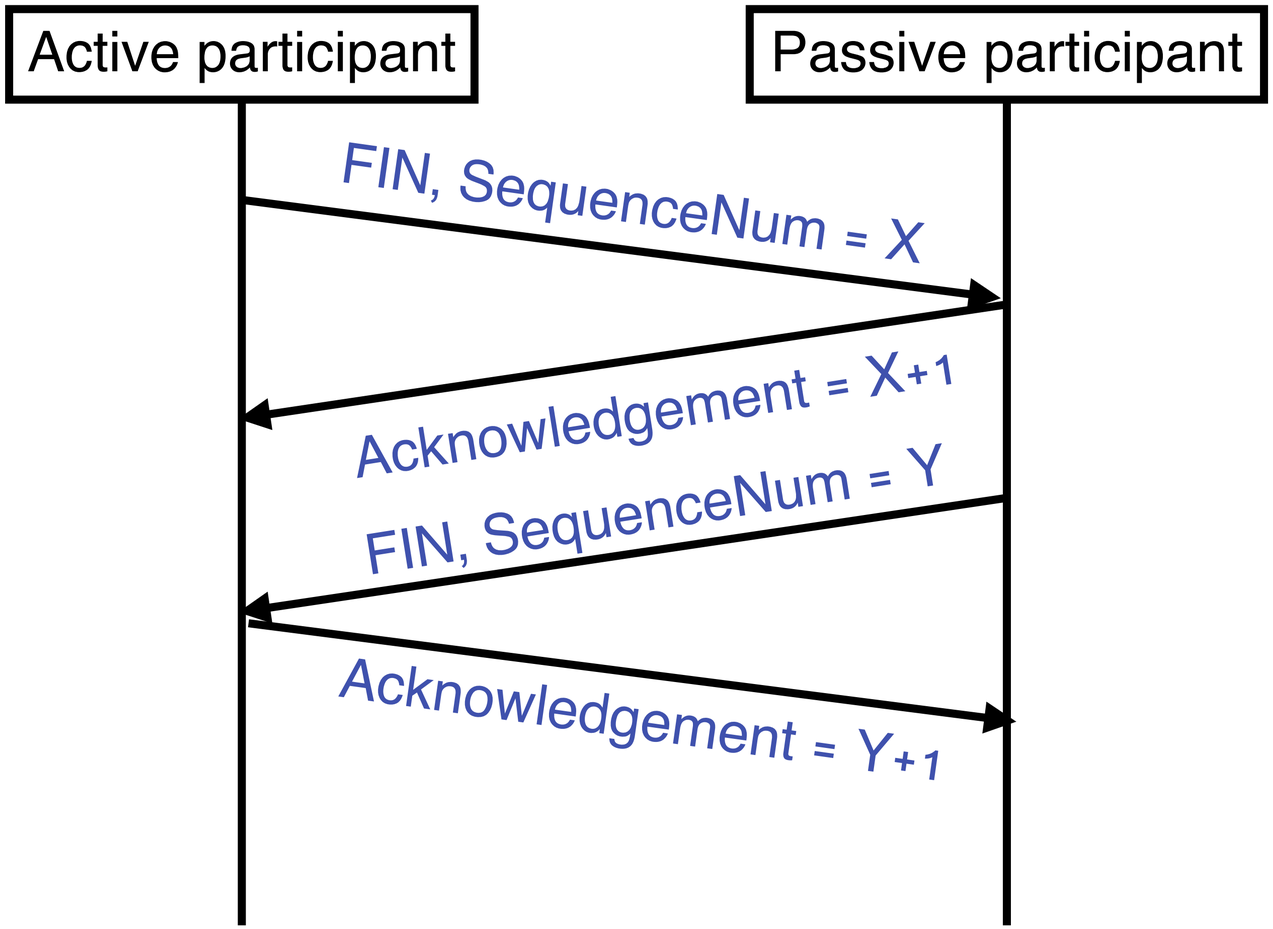


TCP State Transition Diagram

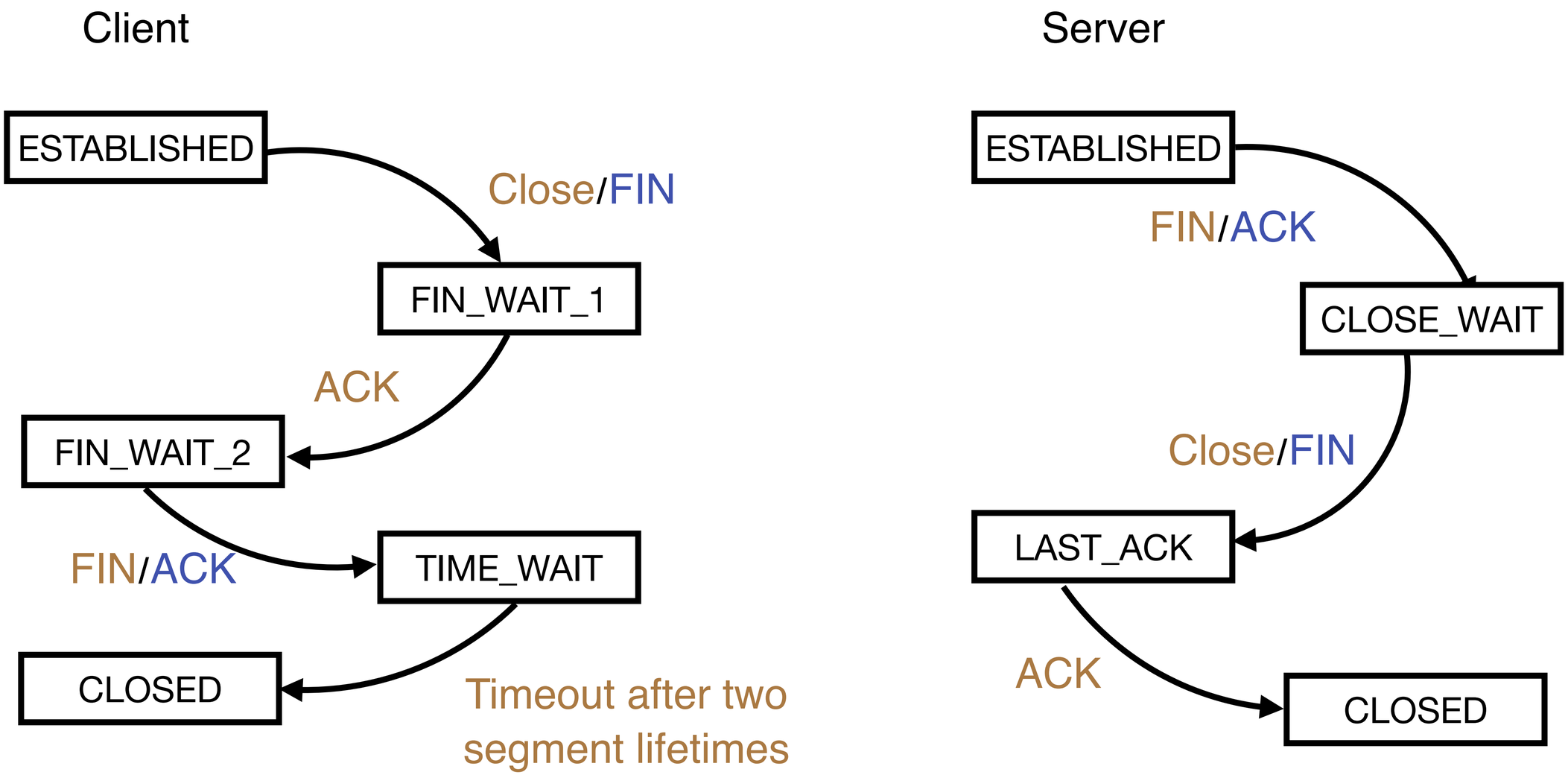


TCP Connection Teardown

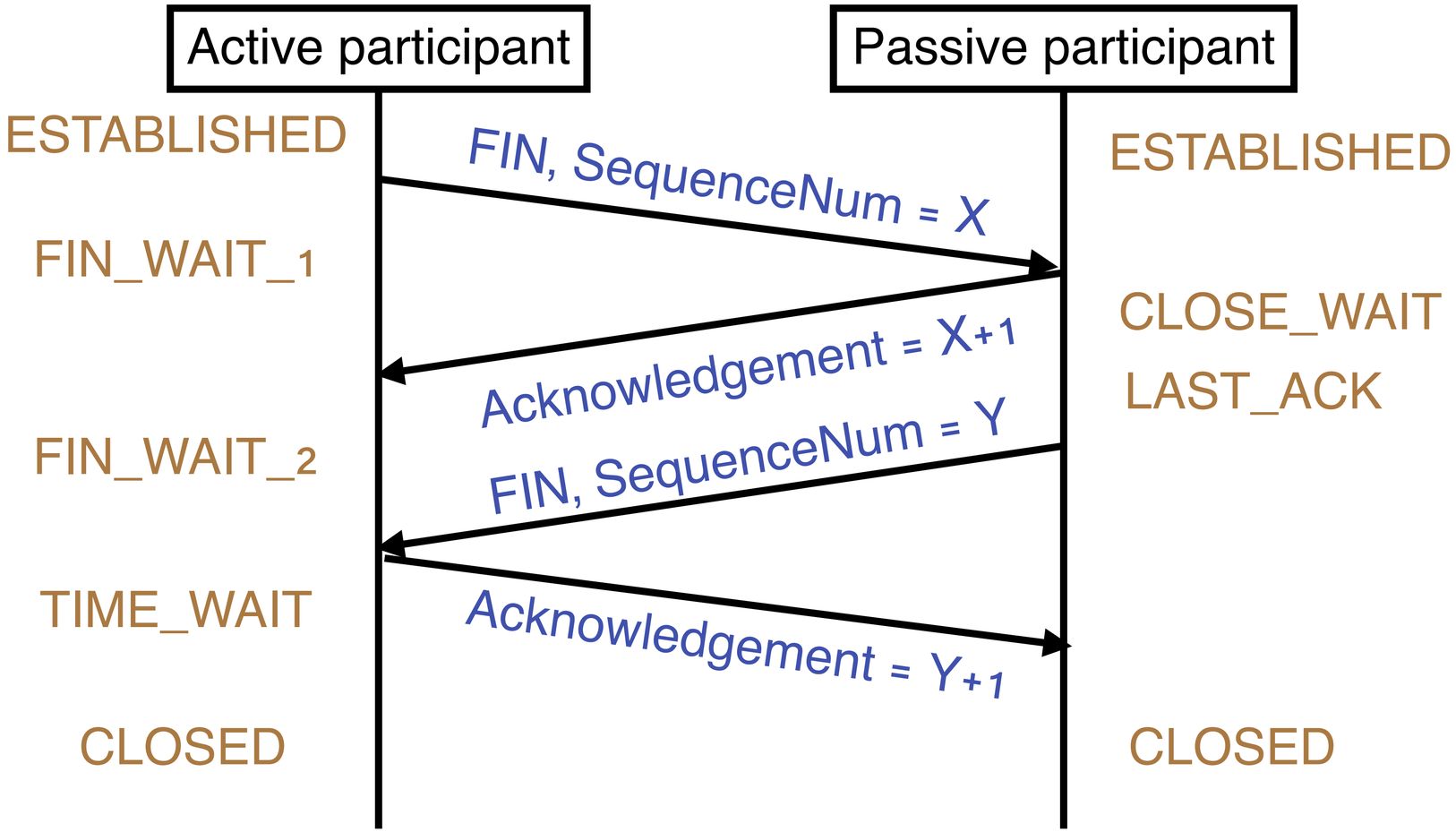
- Case 1: 4-way handshake



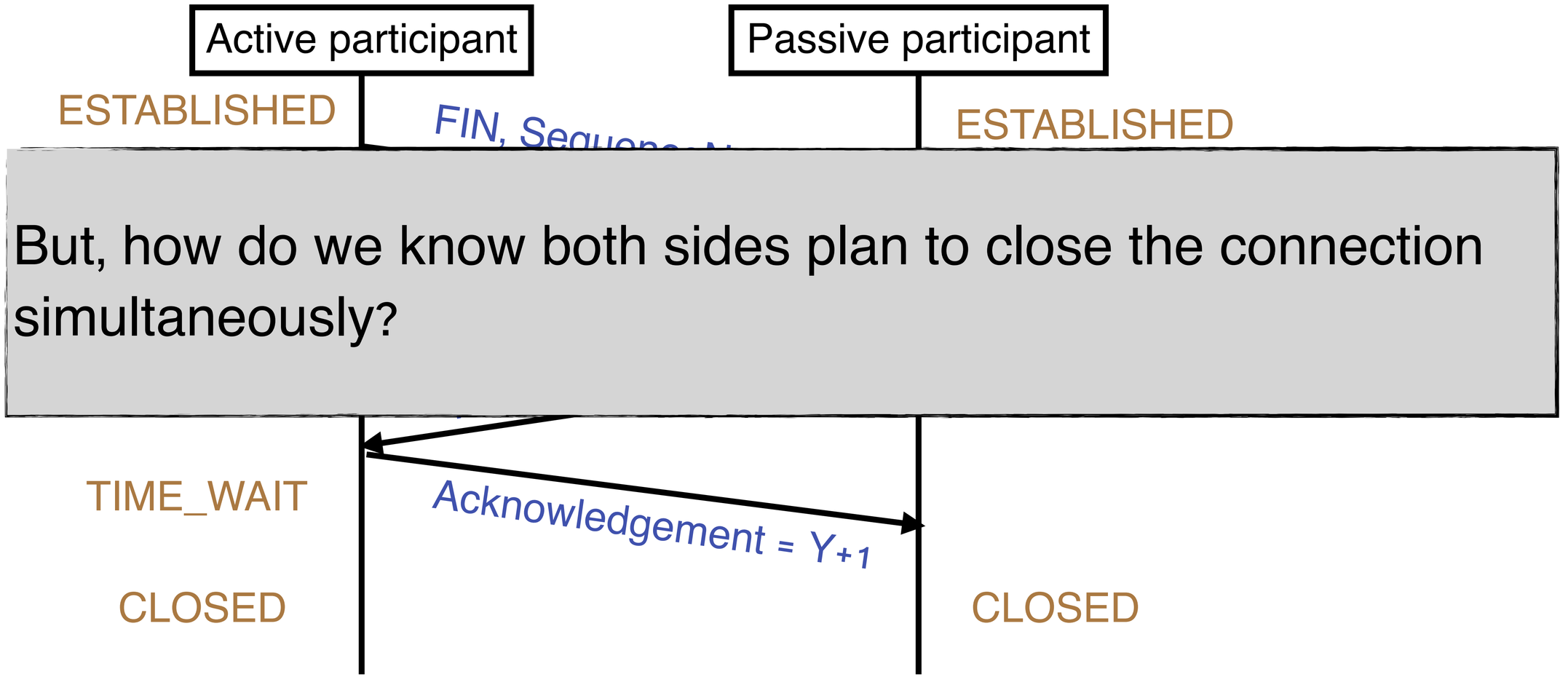
TCP State Machine Transition



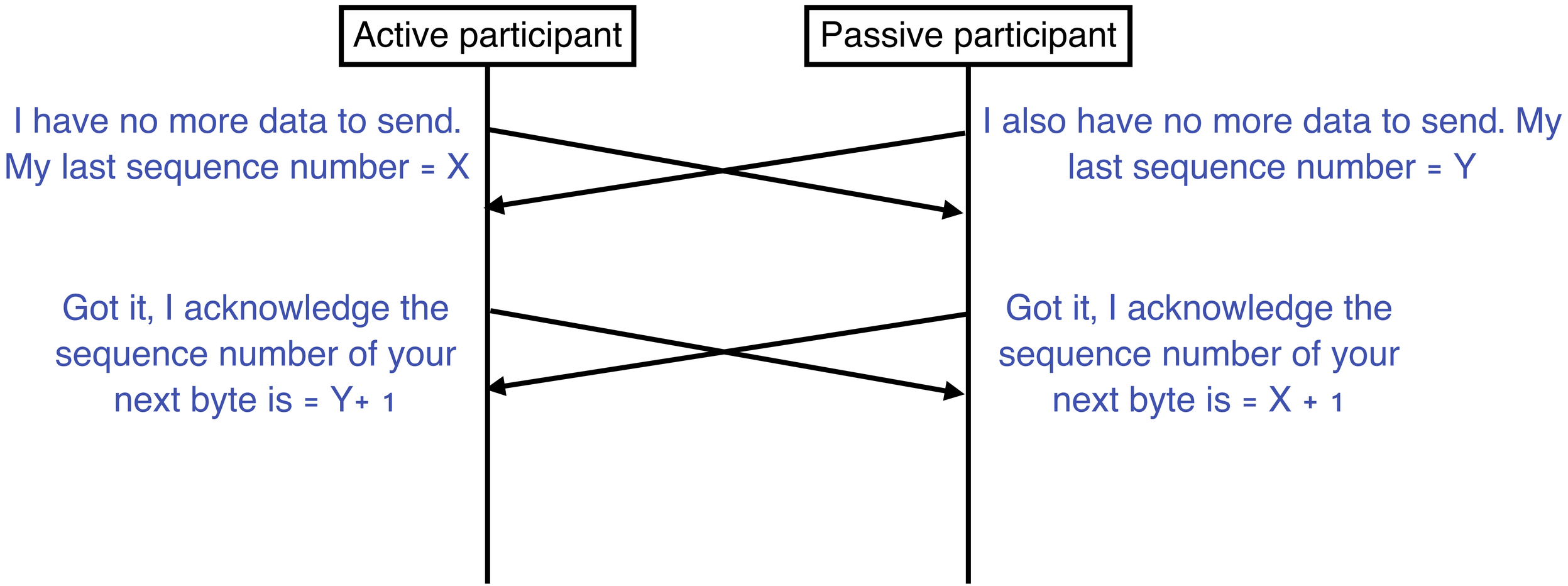
TCP Connection Termination (Case 1)



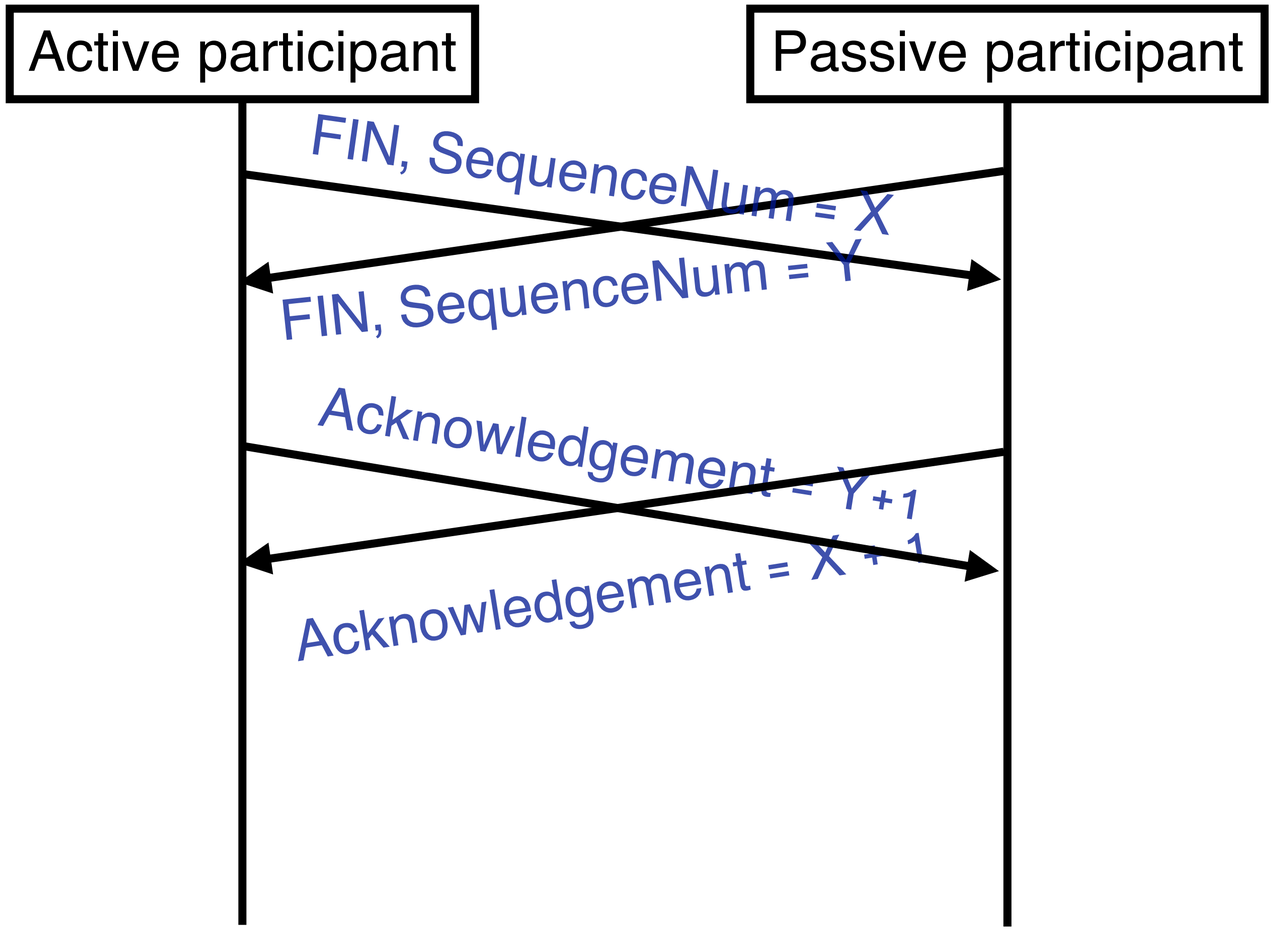
TCP Connection Termination (Case 1)



Case 2: Both Slides Close Simultaneously



Case 2: Both Slides Close Simultaneously



Case 2: State Machine Transition (Step 1)

Client

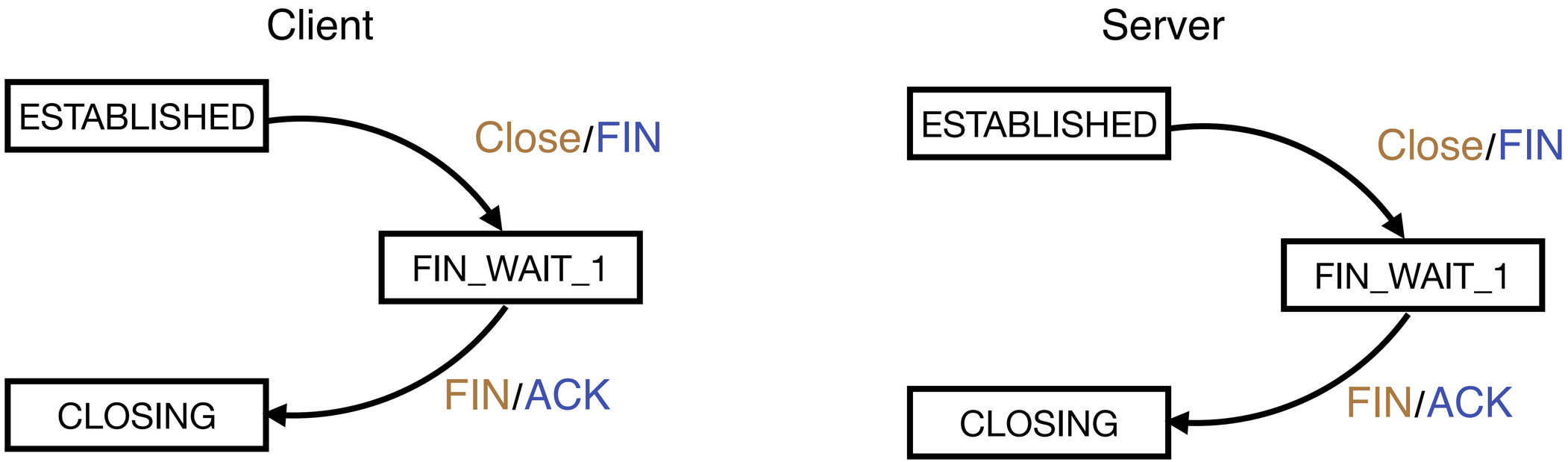
ESTABLISHED

Server

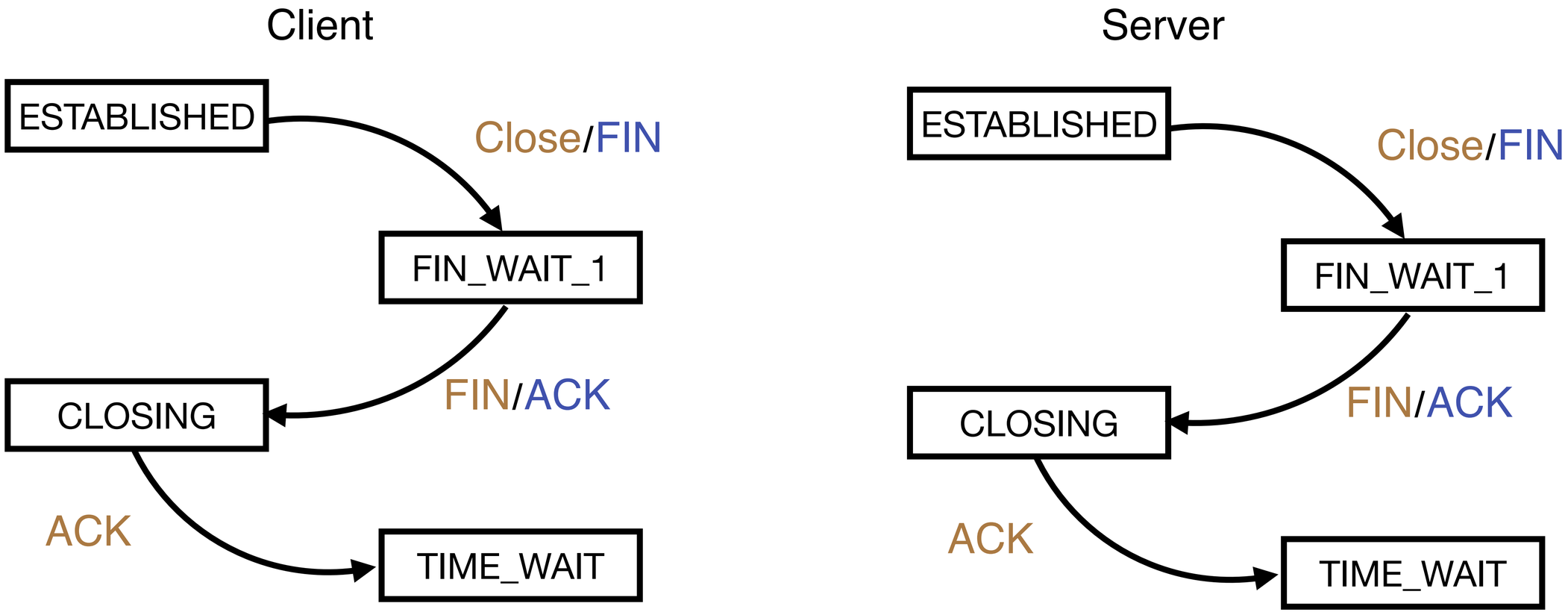
ESTABLISHED

Case 2: State Machine Transition (Step 1)

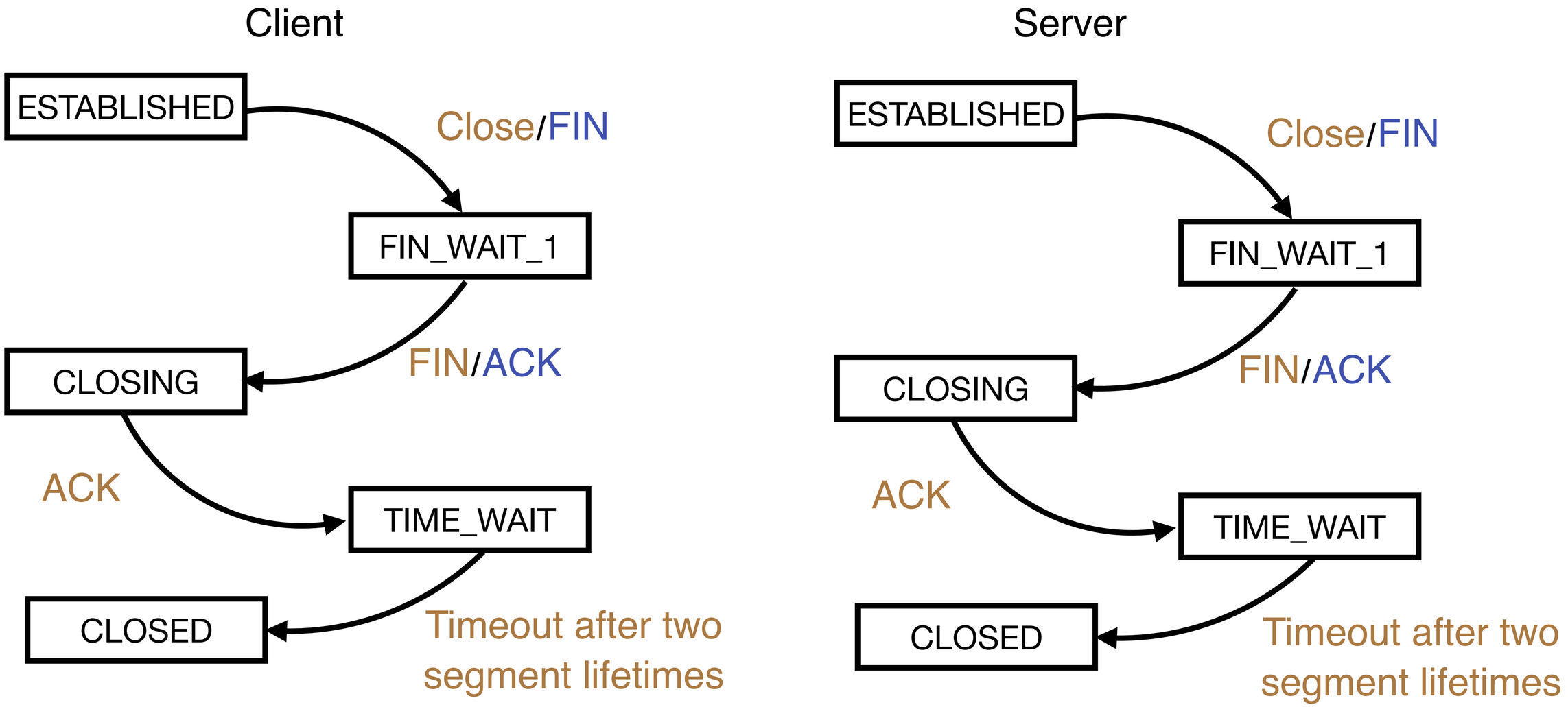
Case 2: State Machine Transition (Step 2)



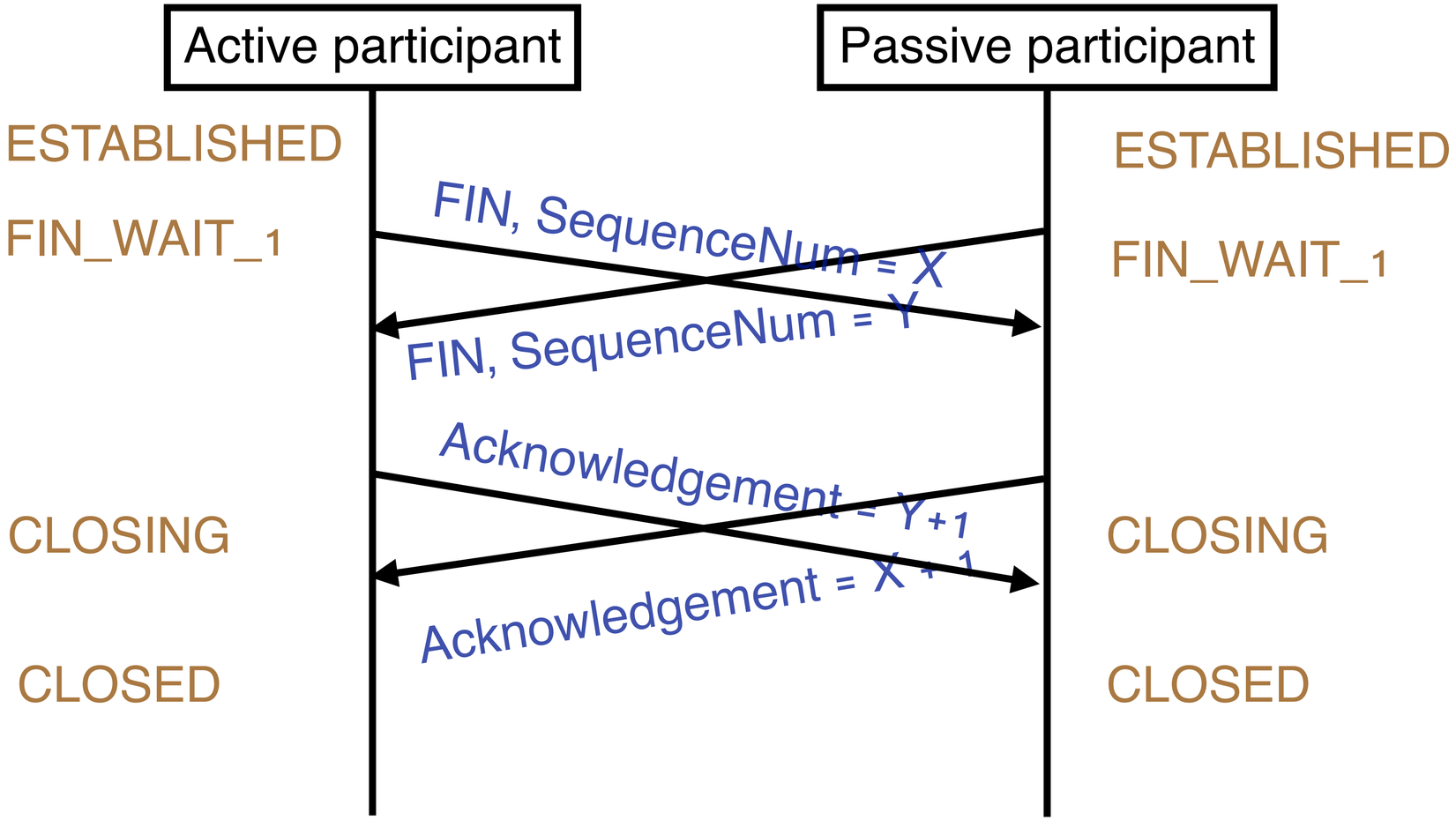
Case 2: State Machine Transition (Step 3)



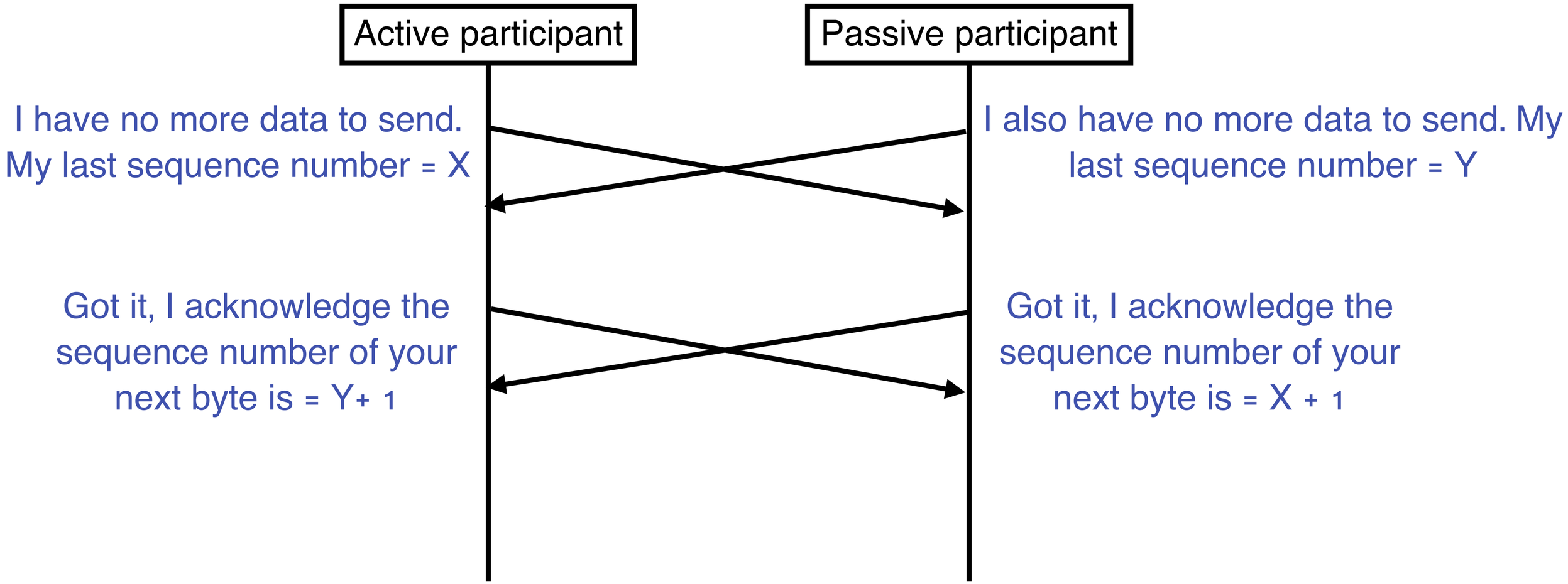
Case 2: State Machine Transition (Step 4)



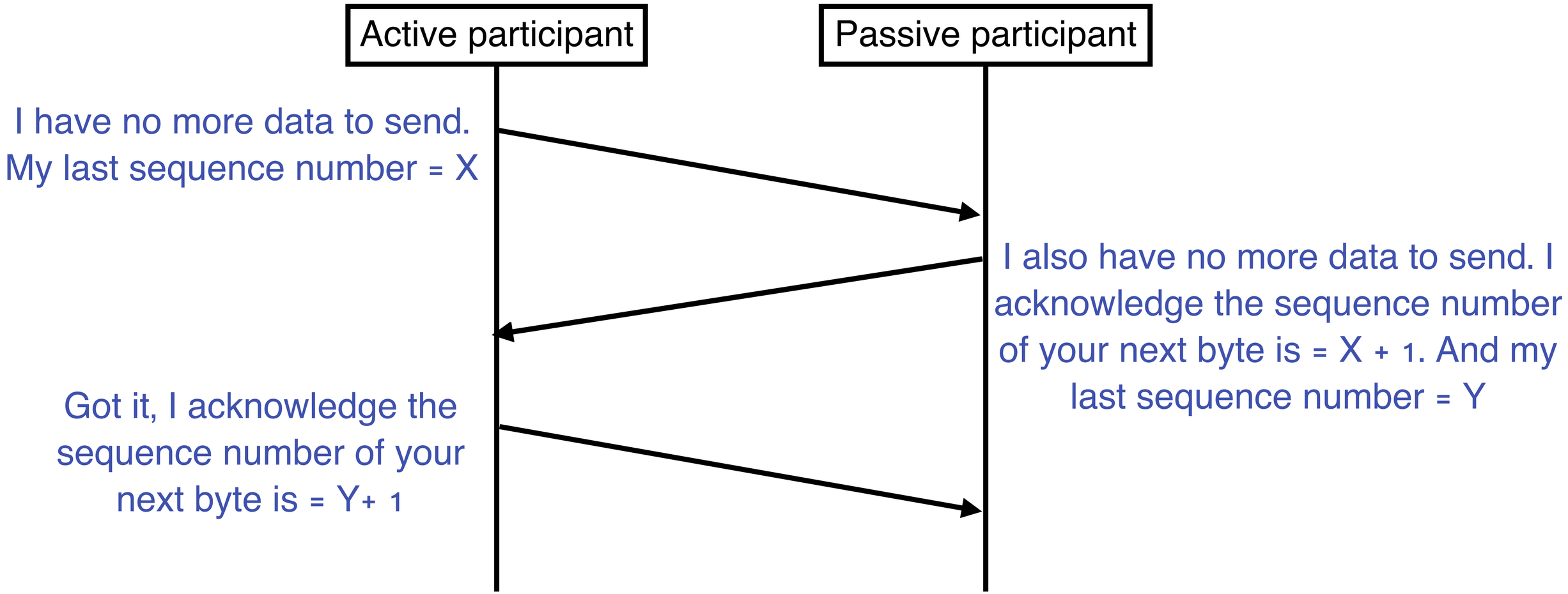
TCP Connection Termination (Case 2) Summary



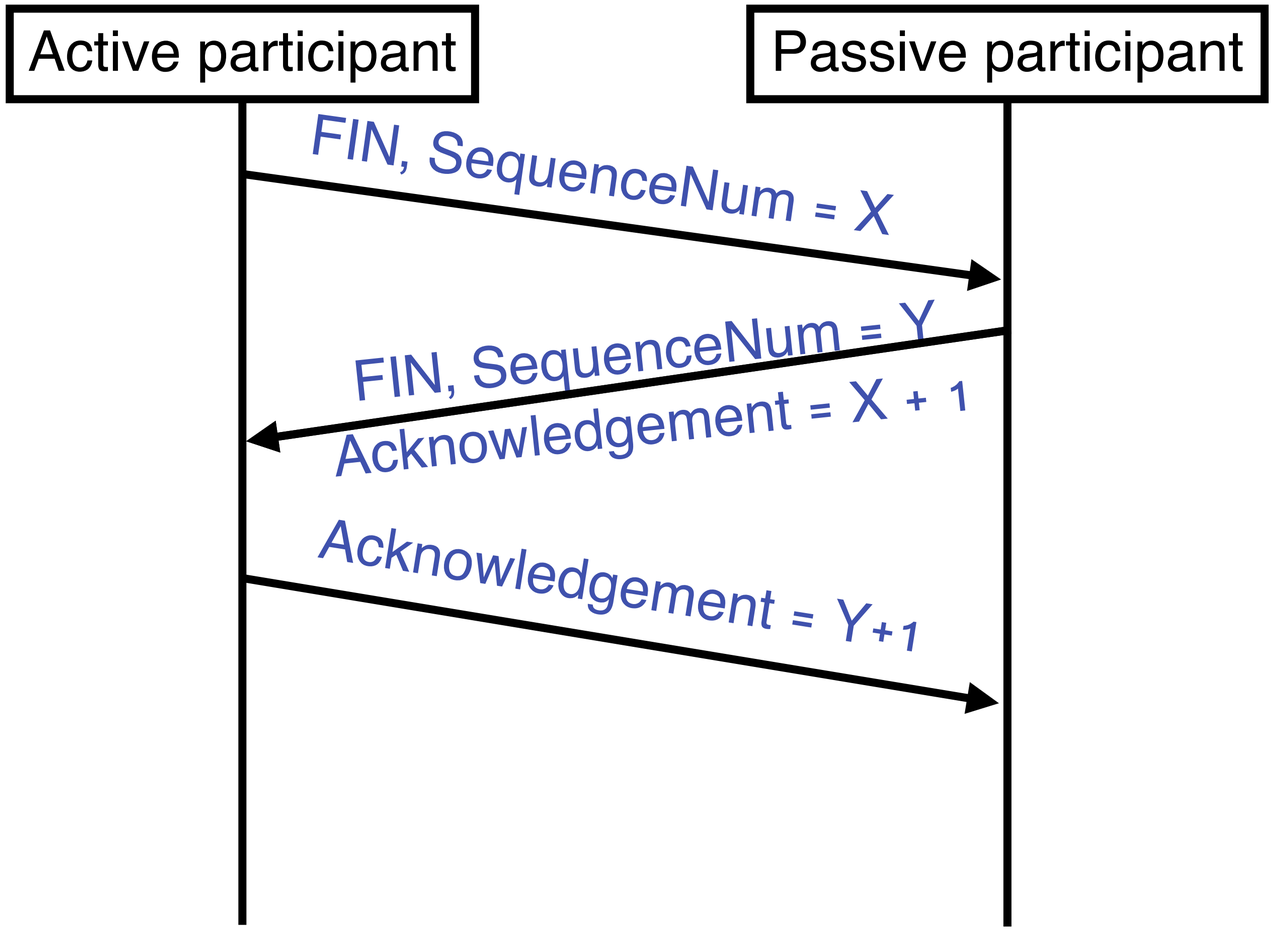
Case 3: Both Sides Close Simultaneously, but



Case 3: Message Order Changes



TCP Connection Termination (Case 3)



Case 3: State Machine Transition (Step 1)

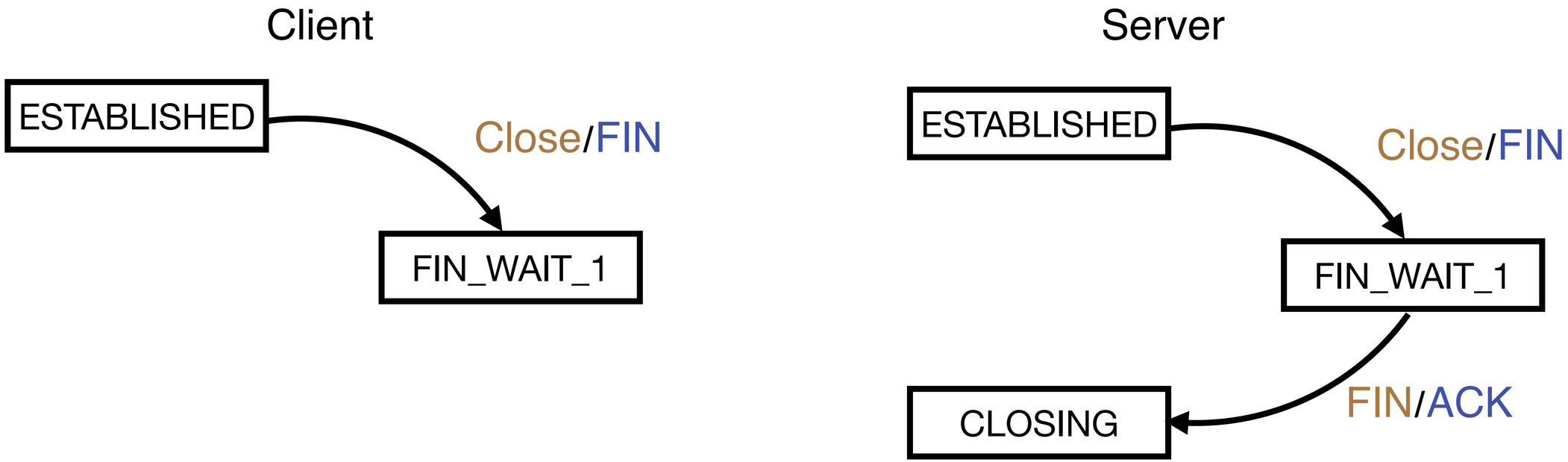
Client

ESTABLISHED

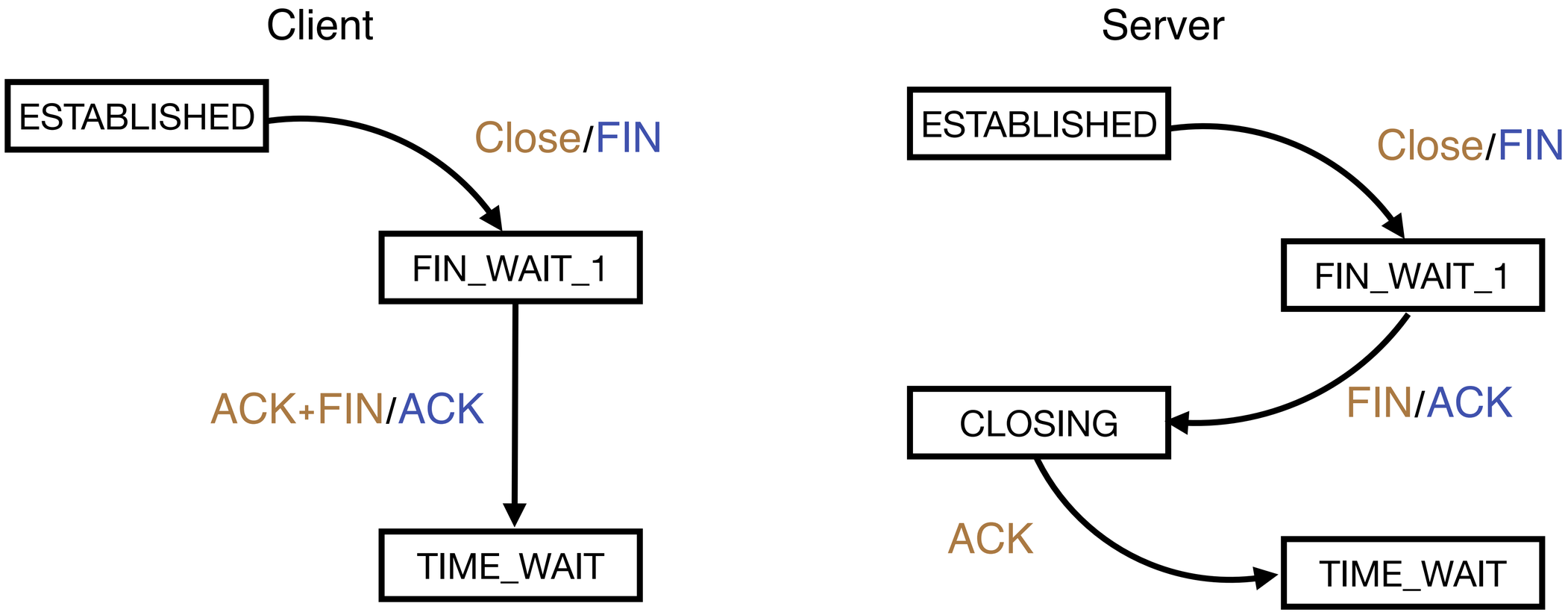
Server

ESTABLISHED

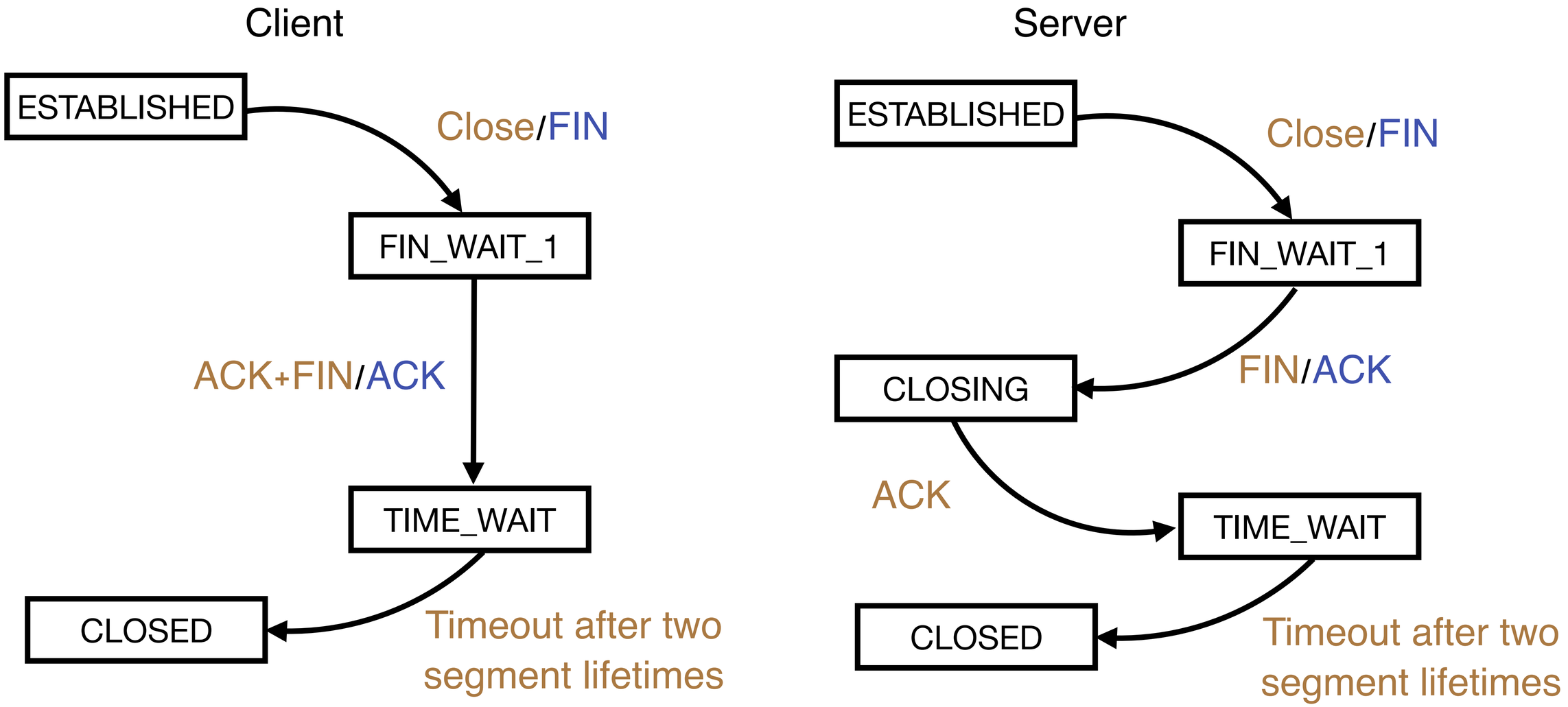
Case 3: State Machine Transition (Step 1)



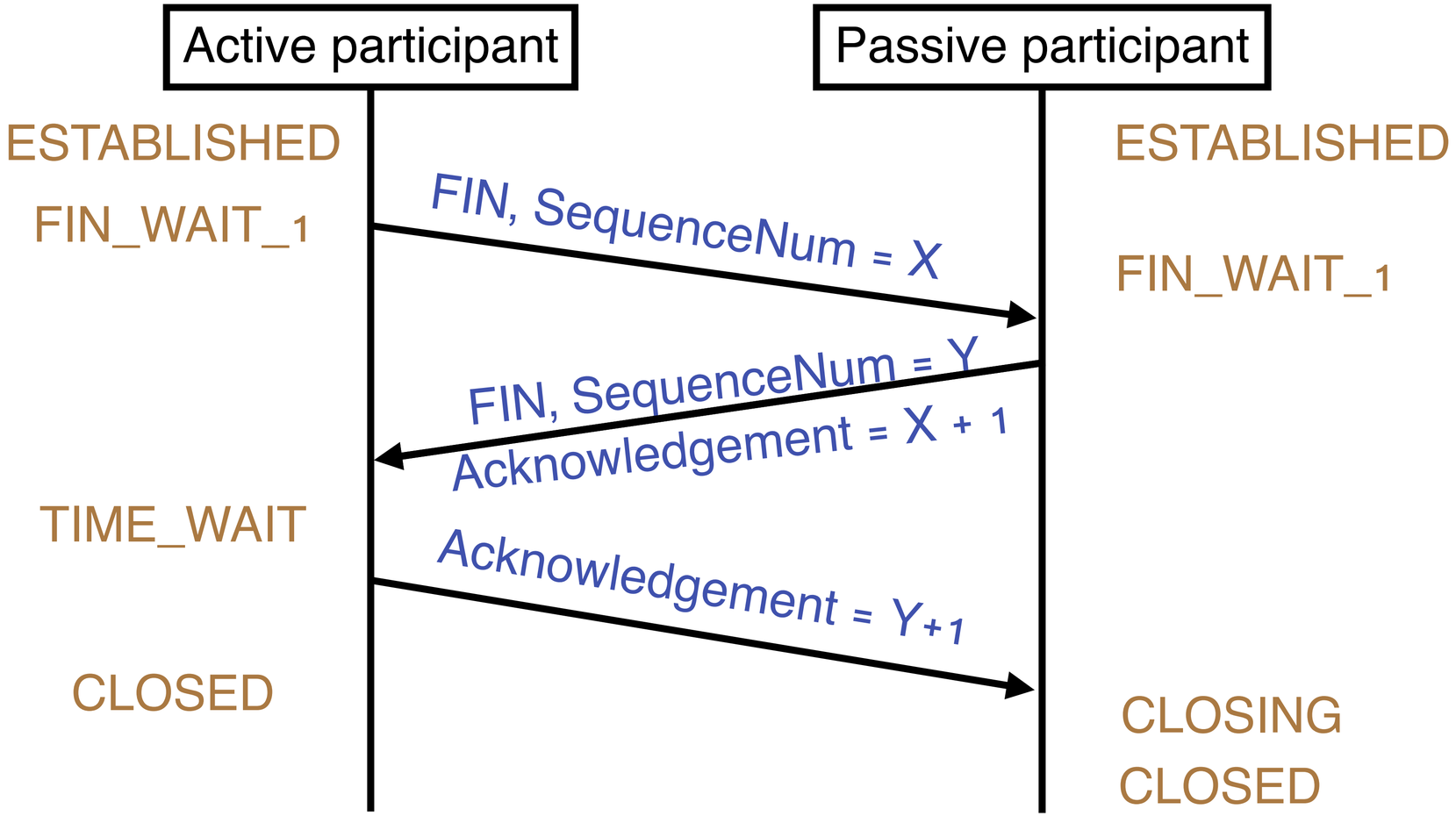
Case 3: State Machine Transition (Step 2)



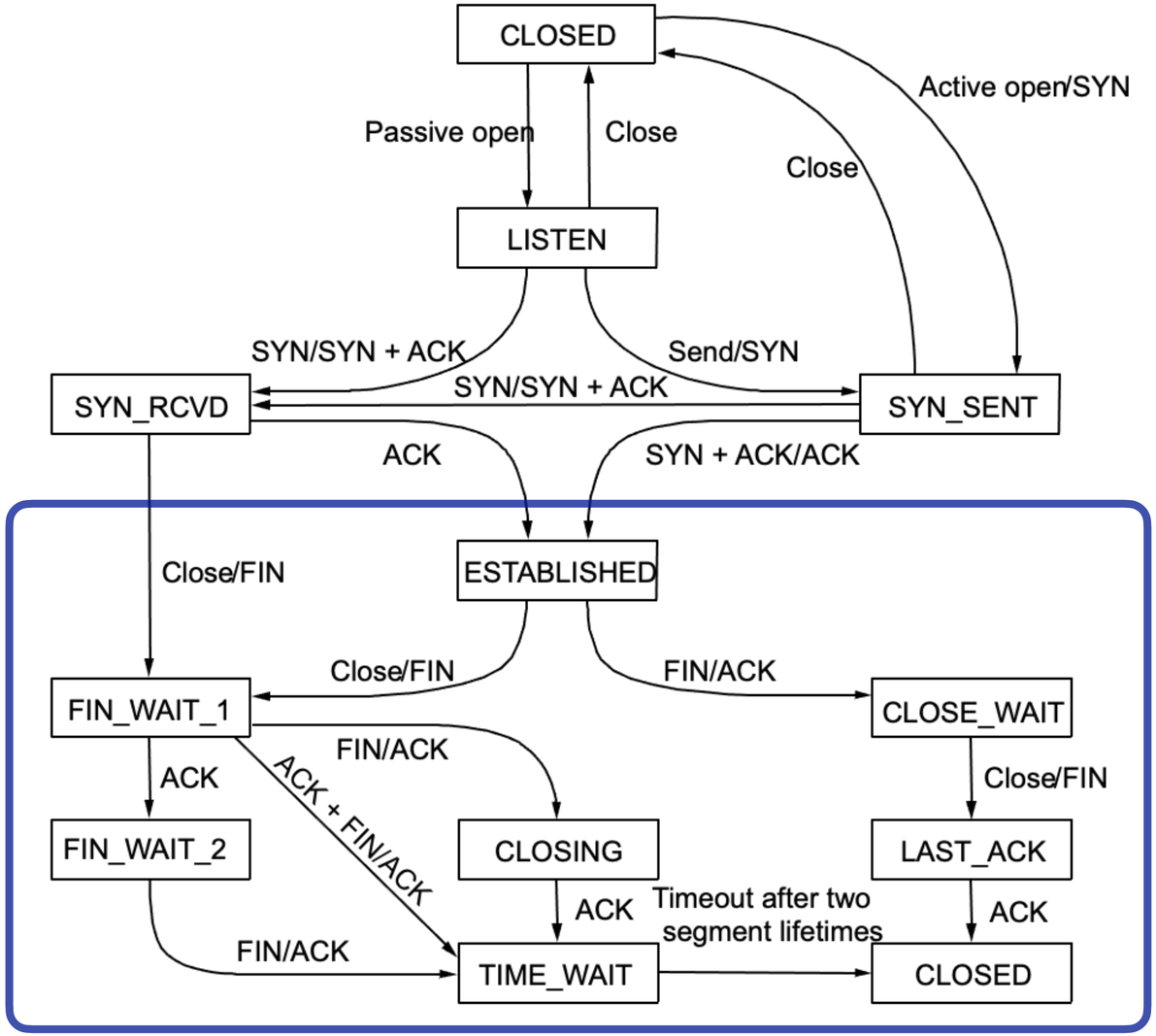
Case 3: State Machine Transition (Step 3)



TCP Connection Termination (Case 3) Summary



TCP State Transition Diagram



TCP Connection Management Summary

- #1: Connection setup is asymmetric
  - One side does a passive open and 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 and the other side does an active open
- #2: Connection teardown is symmetric
  - Each side has to close the connection independently
- #3: Most states schedule a timeout
  - Timeouts are triggered when the expected responses does not happen

TCP Connection Management Summary

- #1: Connection setup is asymmetric
  - One side does a passive open and 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 does TCP solve 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
  - A checksum is not enough

- #3: No resource management
  - Each channel works as an exclusive network resource owner
  - No adaptive support for the physical networks and applications

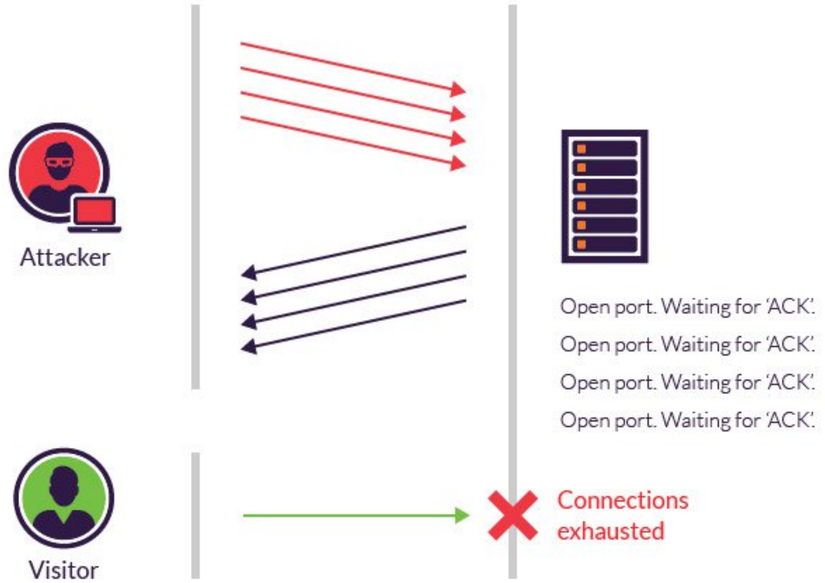
TCP avoids arbitrary communication but exposes non-negligible attacking interfaces.

SYN Flood Attack

- Expected behavior: 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.

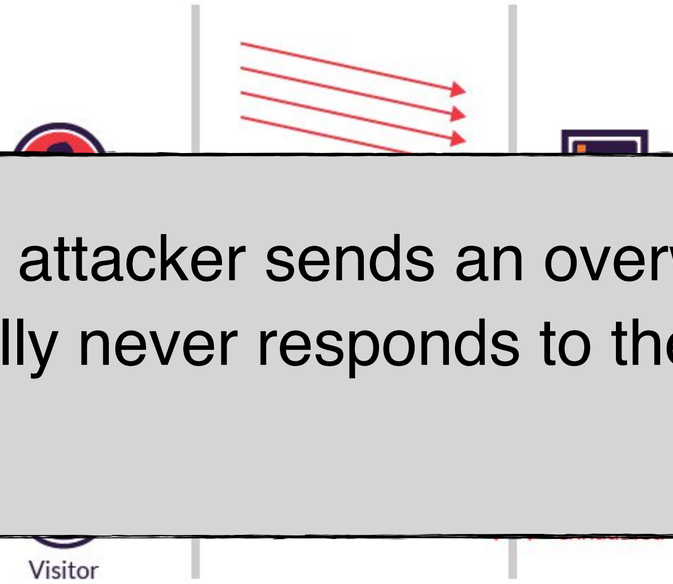
SYN Flood Attack

- Expected behavior: 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 Attack

- Expected behavior: 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.



- Abnormal behavior: An attacker sends an overwhelming number of SYN requests and intentionally never responds to the server's SYN-ACK messages.

Summary

- Today
  - TCP connection management (II)

- Next lecture
  - TCP reliability support (I)