

Introduction to Computer Networks

# **TCP Connection Management (I)**

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

**Ming Liu**

**mgliu@cs.wisc.edu**

# Today

## Last lecture

- What functionalities does the transport layer provide?

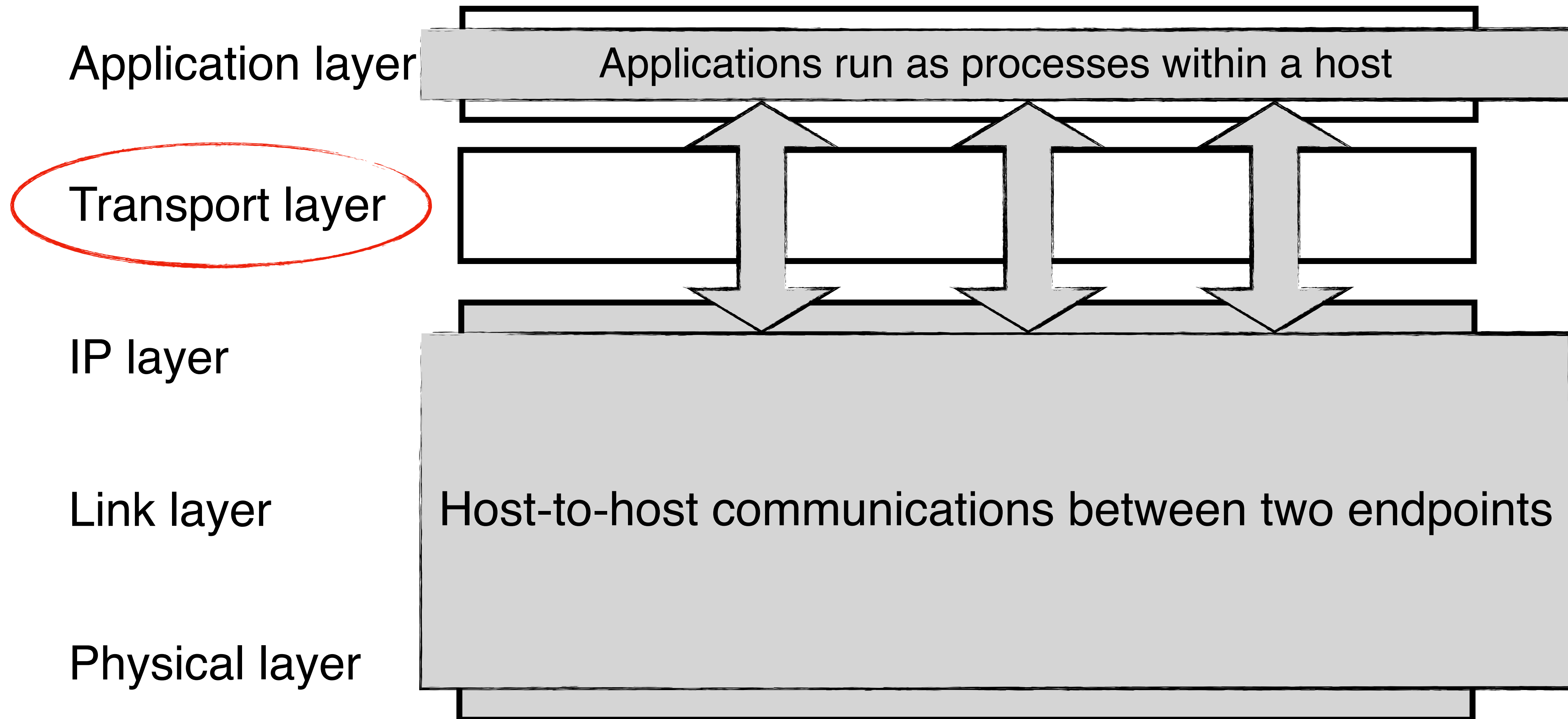
## Today

- How to setup the TCP connection?

## Announcements

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

# Transport Layer in the TCP/IP Model



**Q: What functionalities does the transport layer provide?**

**A: Process-to-process communication channels**

Q1: How to set up the process-to-process channel?

Q2: How to multiplex concurrent channels over the physical link?

Q3: How to control the transmission rate?

Q4: How to achieve reliable delivery?

Q5: How to share the in-network bandwidth resources?

**Q: What functionalities does the transport layer provide?**

**A: Process-to-process communication channels**

**Q1: How to set up the process-to-process channel?**

**Q2: How to multiplex concurrent channels over the physical link?**

**Q3: How to control the transmission rate?**

**Q4: How to achieve reliable delivery?**

**Q5: How to share the in-network bandwidth resources?**

# 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
- 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

**Q: What is the goal of TCP connection management?**

**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**



**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

**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  $\leftrightarrow$  Server

**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**

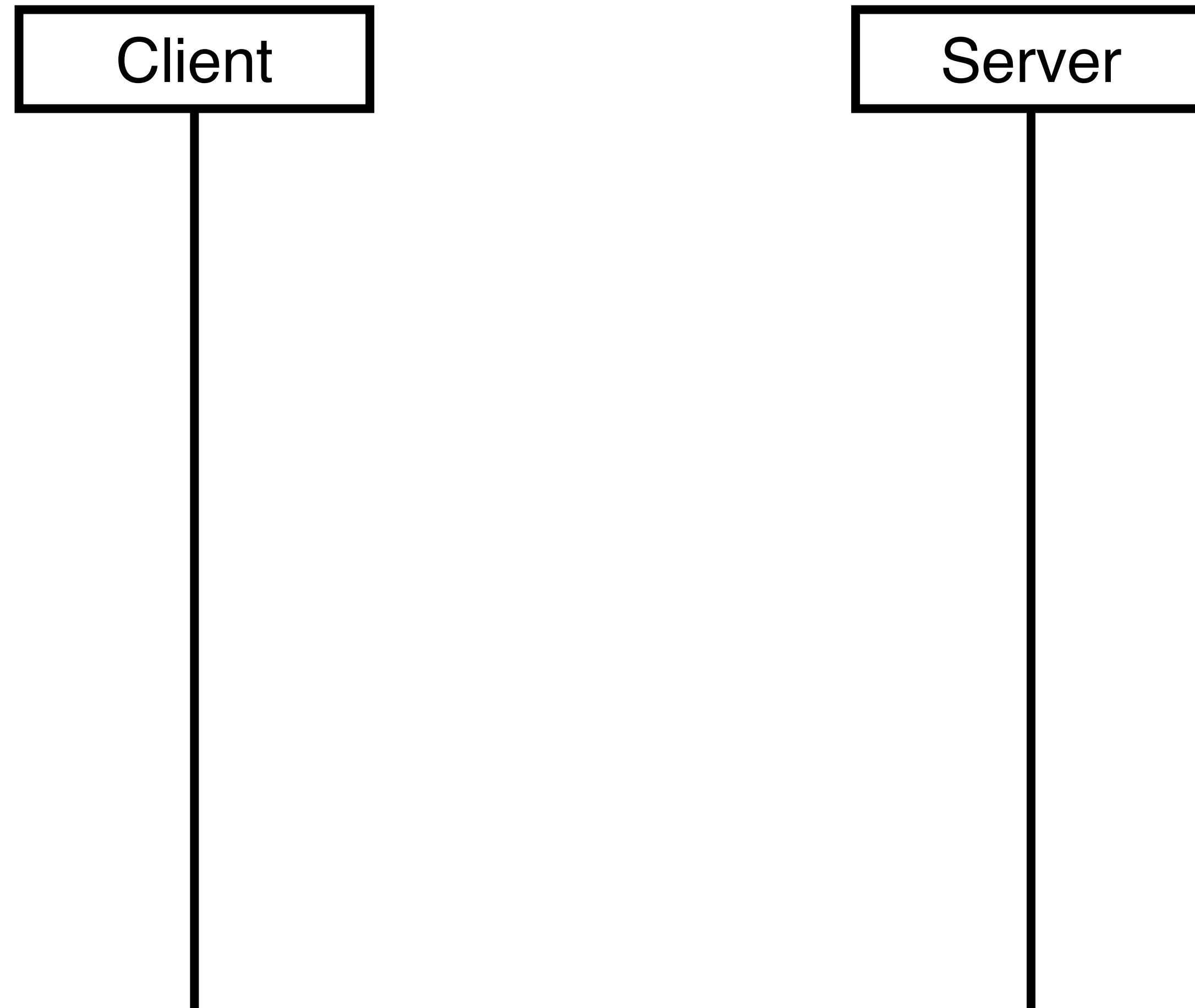
On-demand communication

Client  $\leftrightarrow$  Server

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

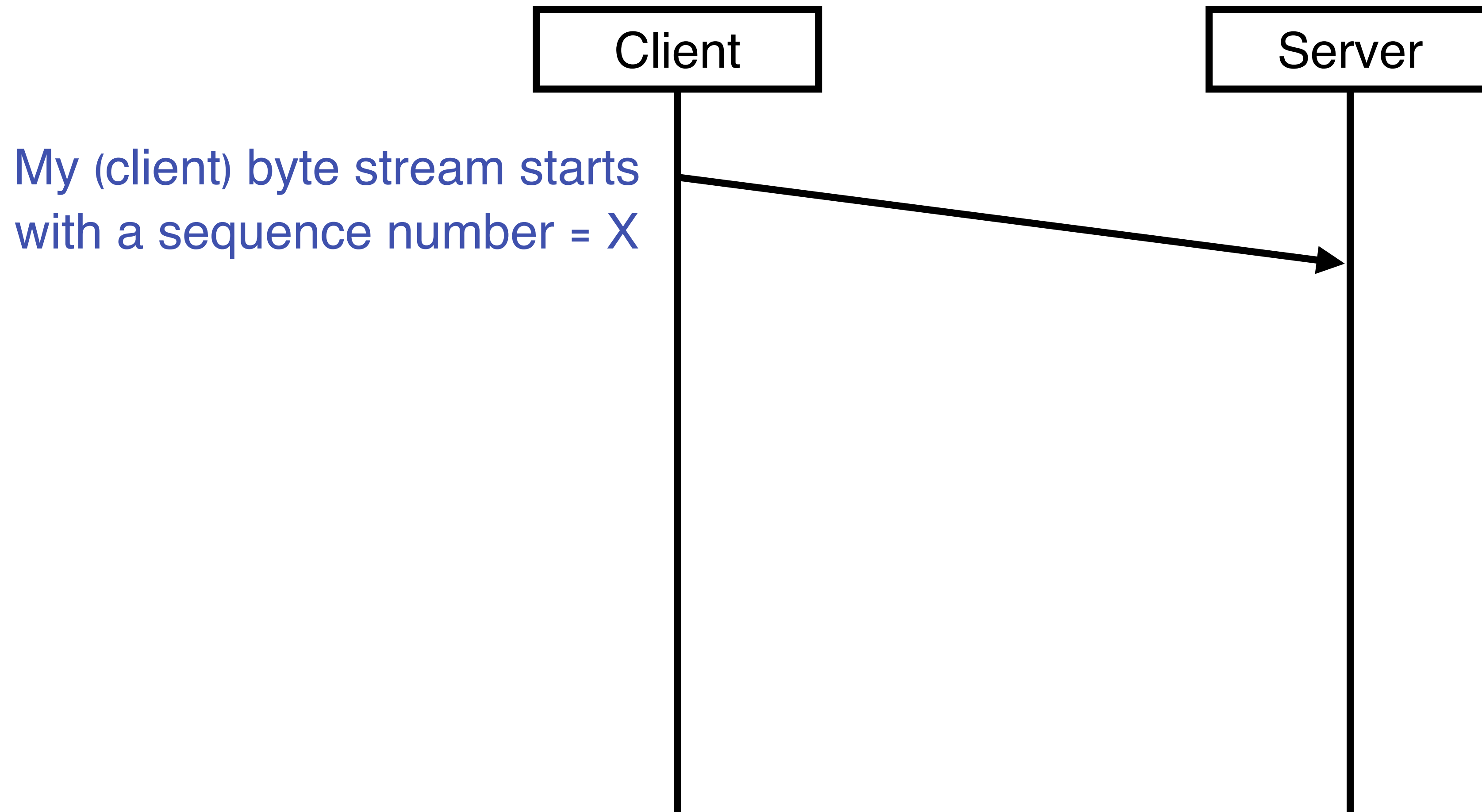
# TCP Connection Establishment

Let's start with a naive approach



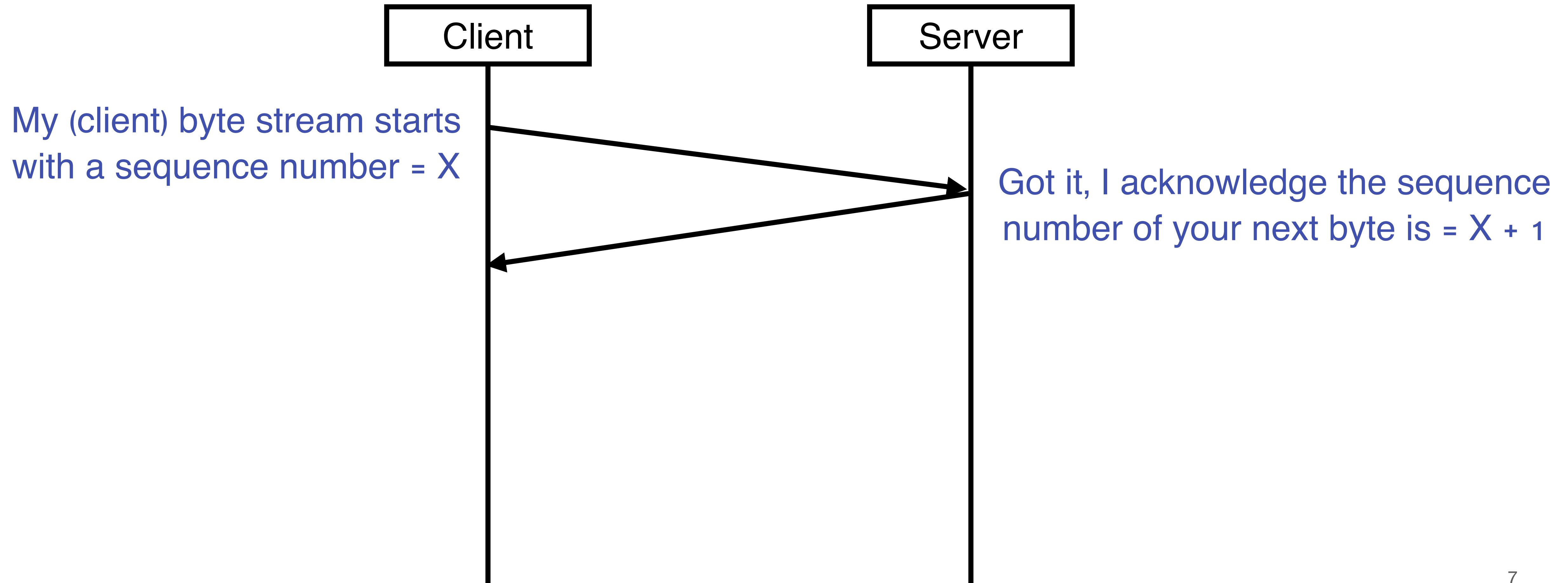
# TCP Connection Establishment

Let's start with a naive approach



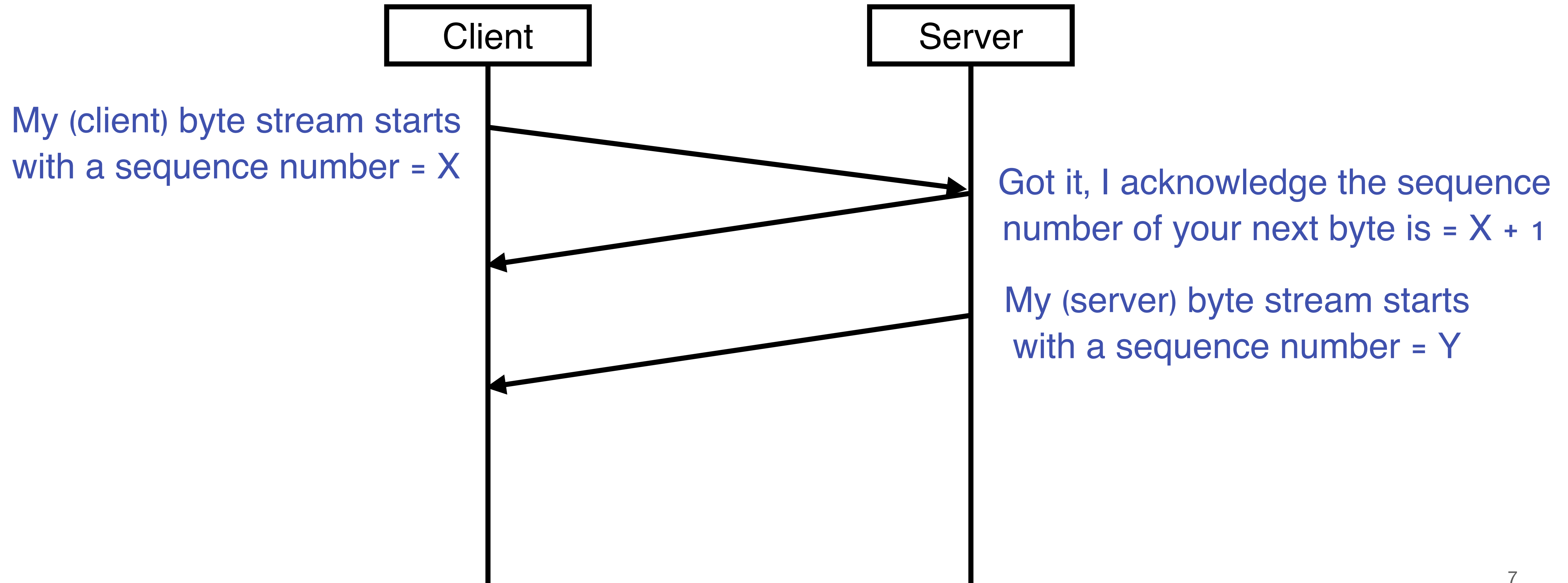
# TCP Connection Establishment

## Let's start with a naive approach



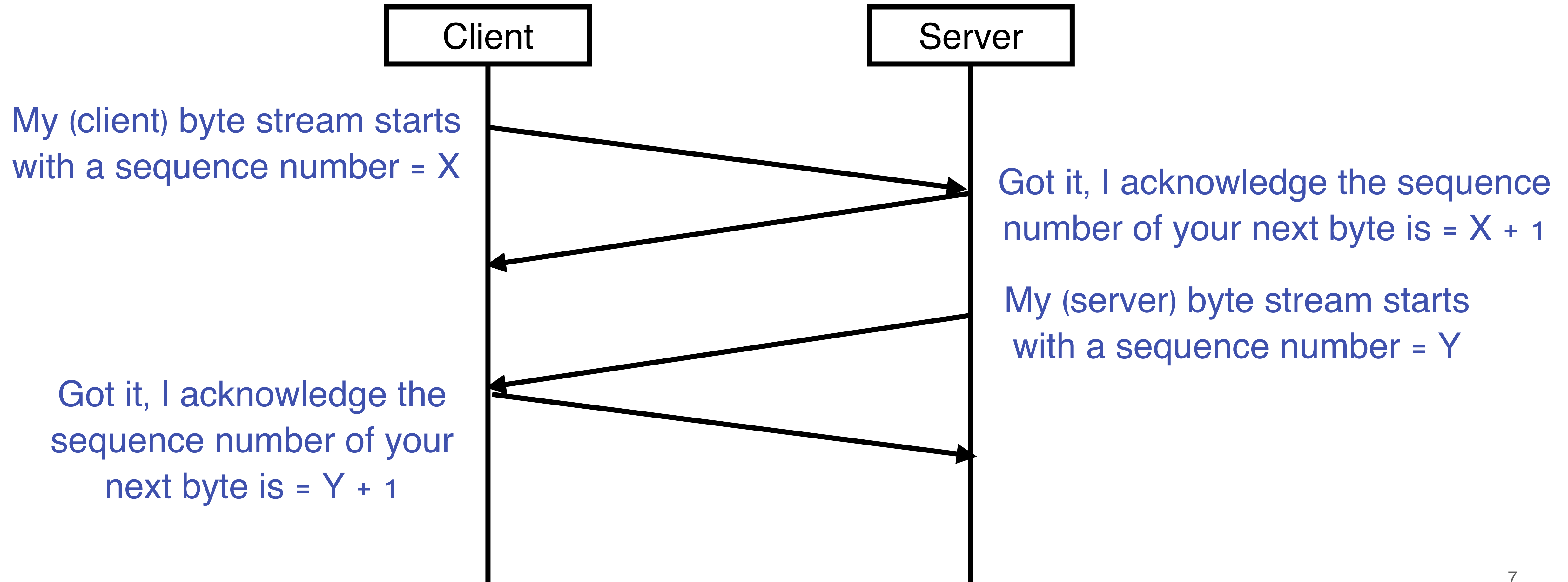
# TCP Connection Establishment

## Let's start with a naive approach



# TCP Connection Establishment

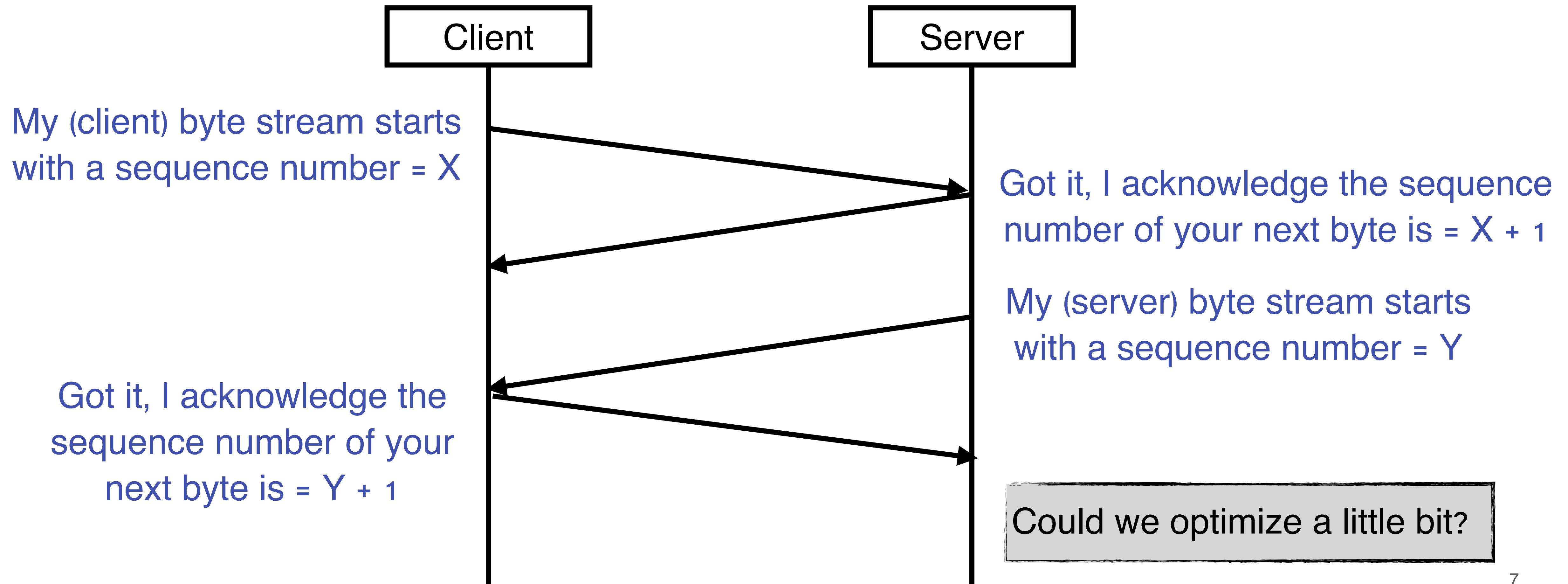
## Let's start with a naive approach





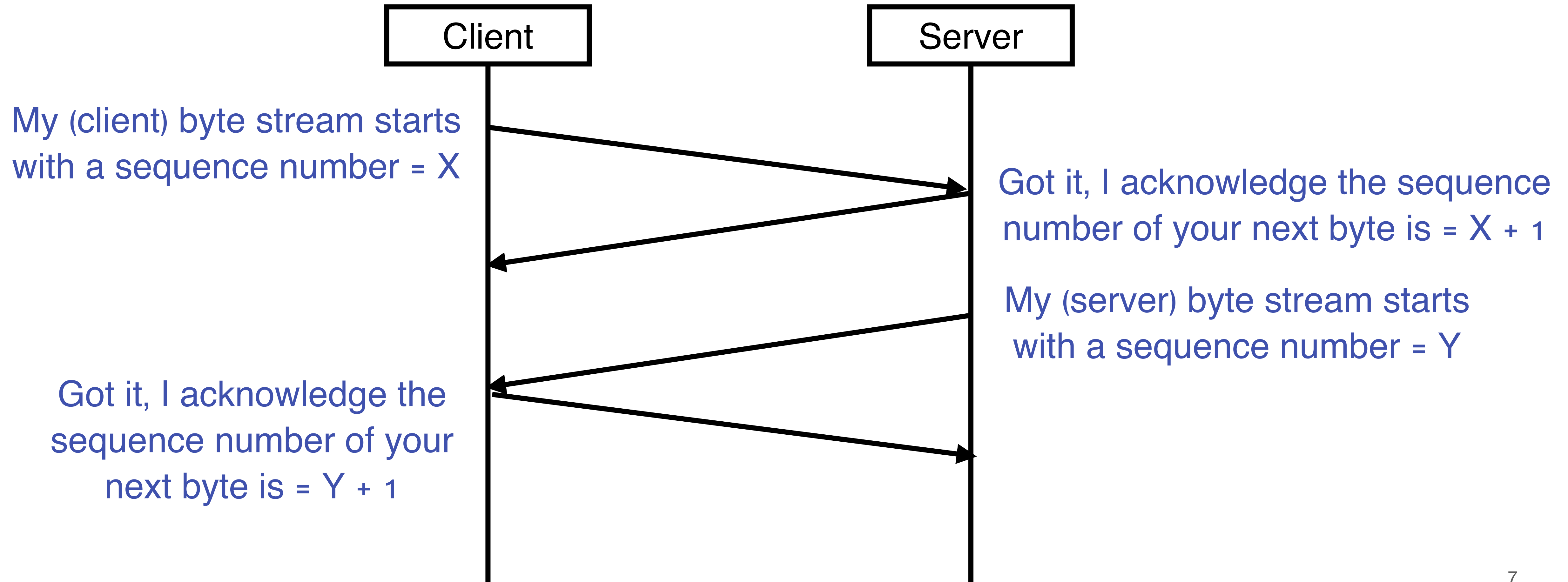
# TCP Connection Establishment

## Let's start with a naive approach



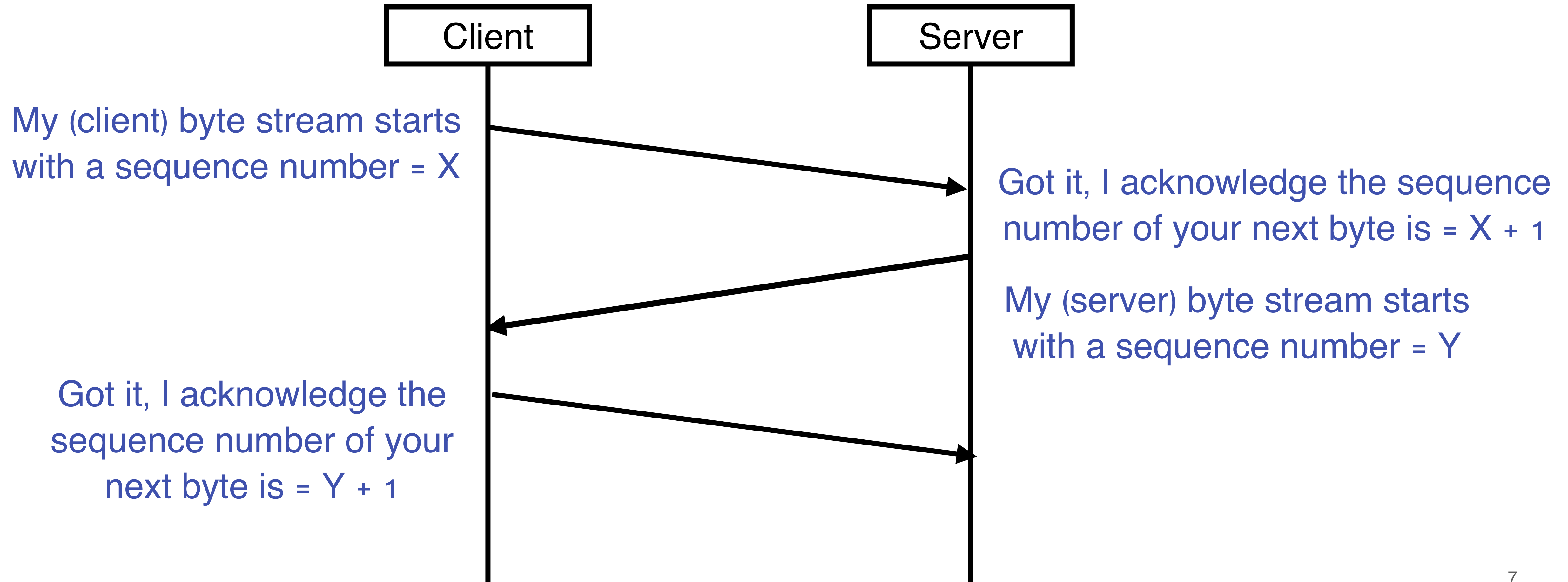
# TCP Connection Establishment

## Let's start with a naive approach

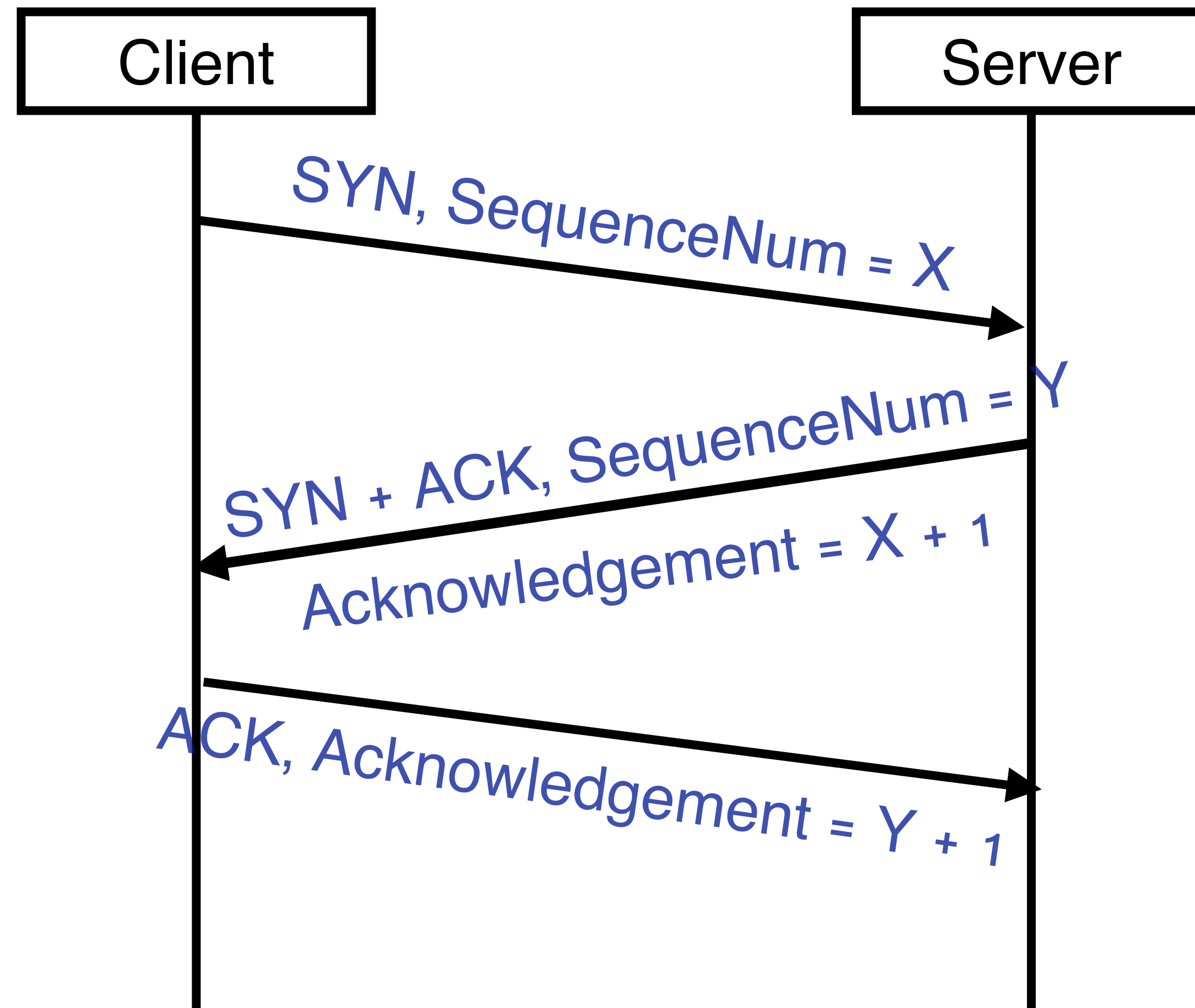


# TCP Connection Establishment

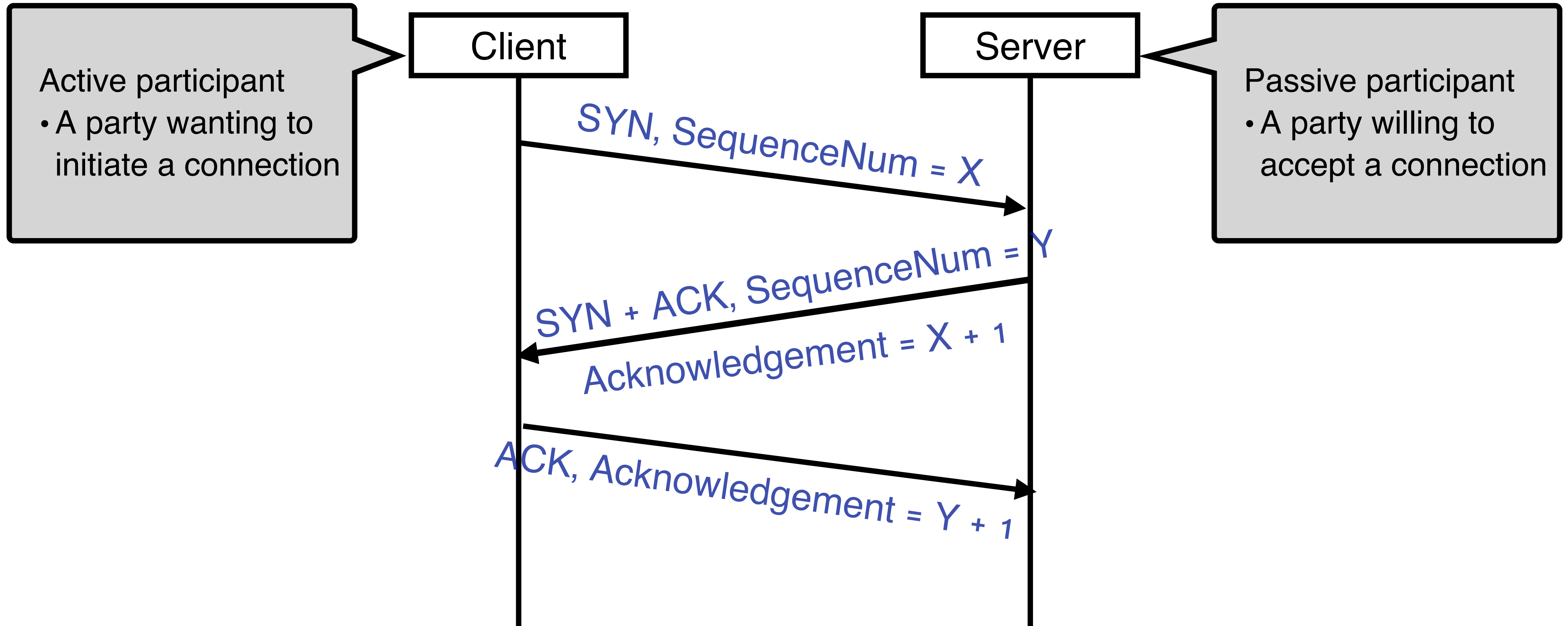
## Let's start with a naive approach



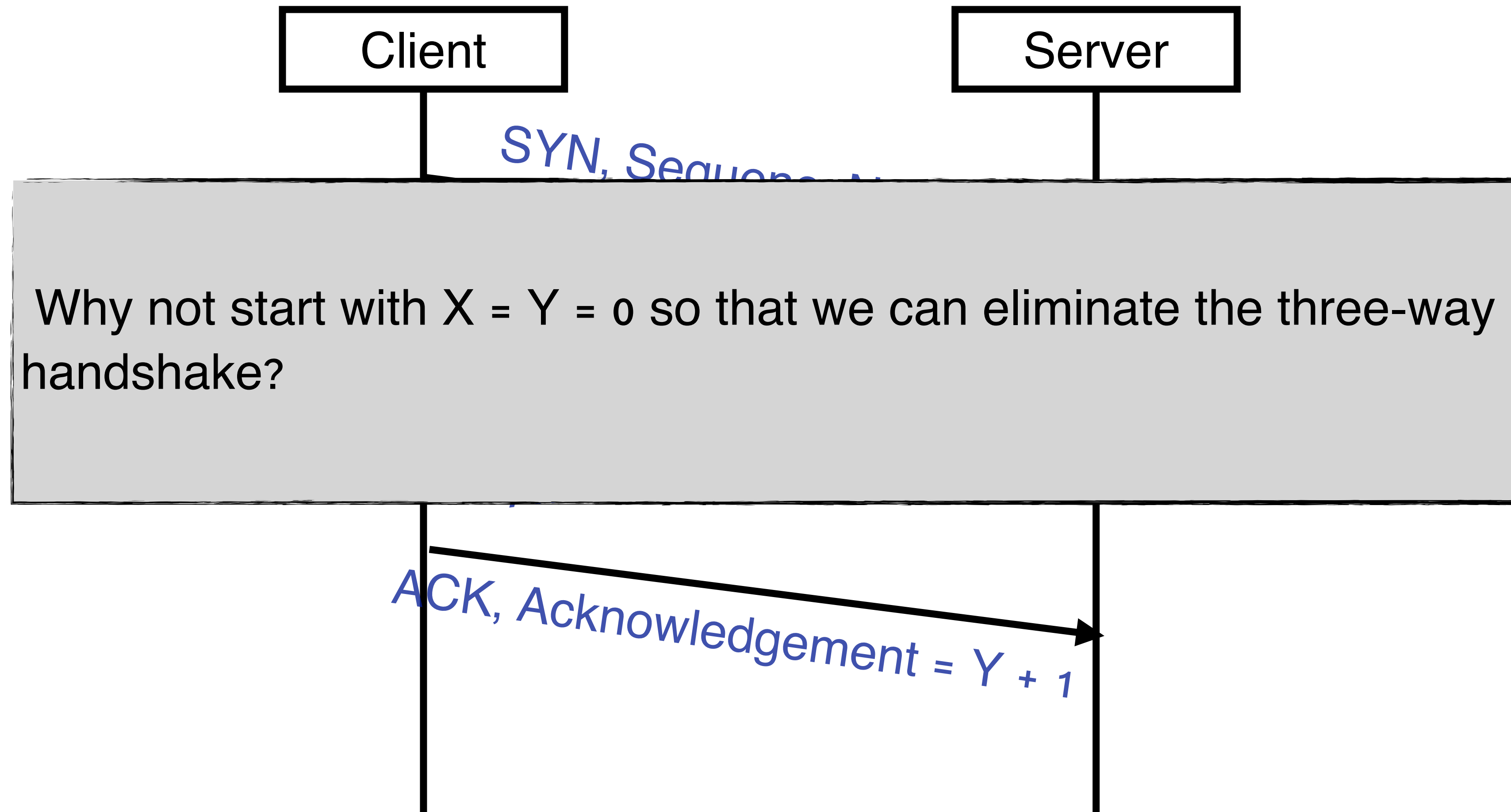
# Three-Way Handshake



# Three-Way Handshake



# Three-Way Handshake

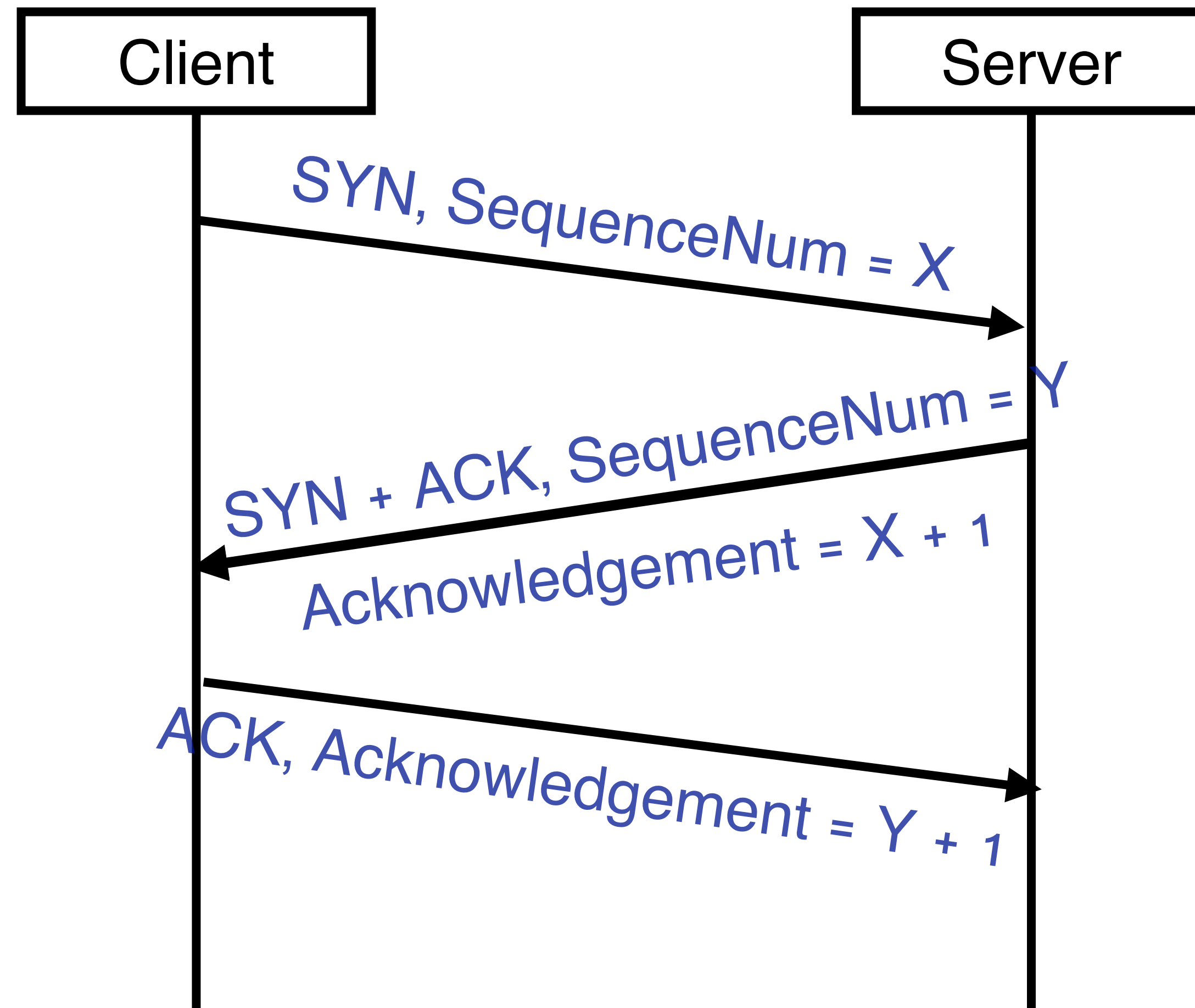


# **The Incarnation Issue**

**A connection (defined by a particular host and port pair) to be reused again**

**Solution: initial sequence number is randomly generated**

# How to implement this?

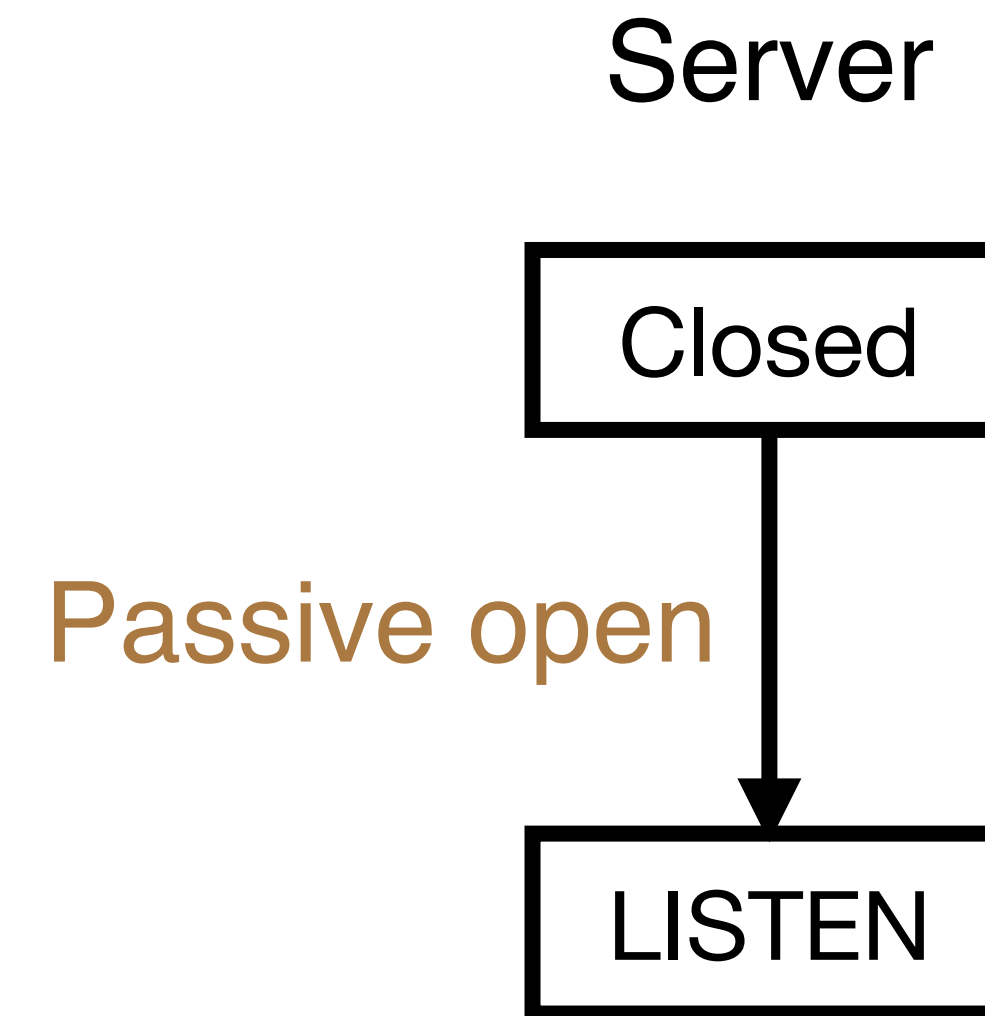
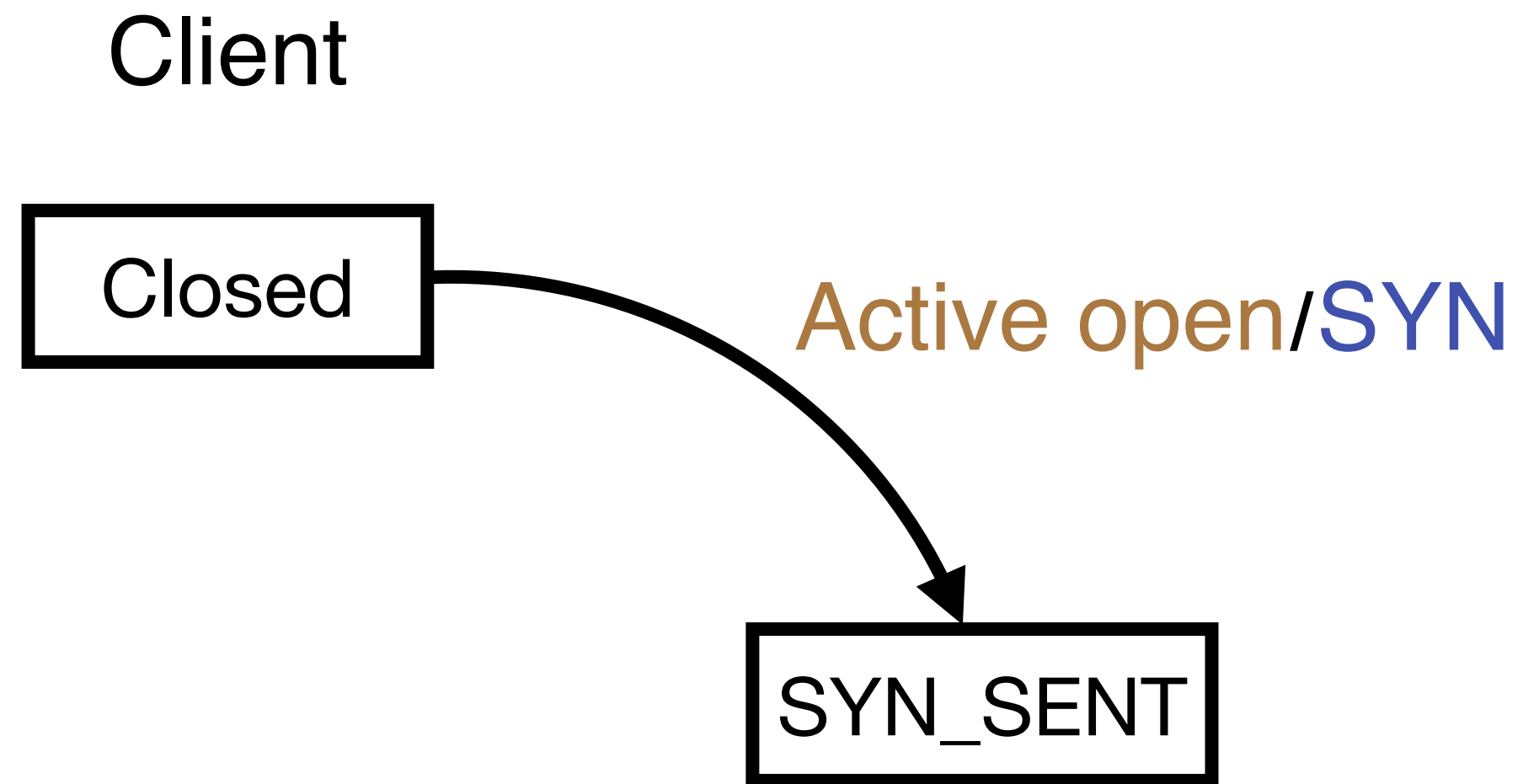




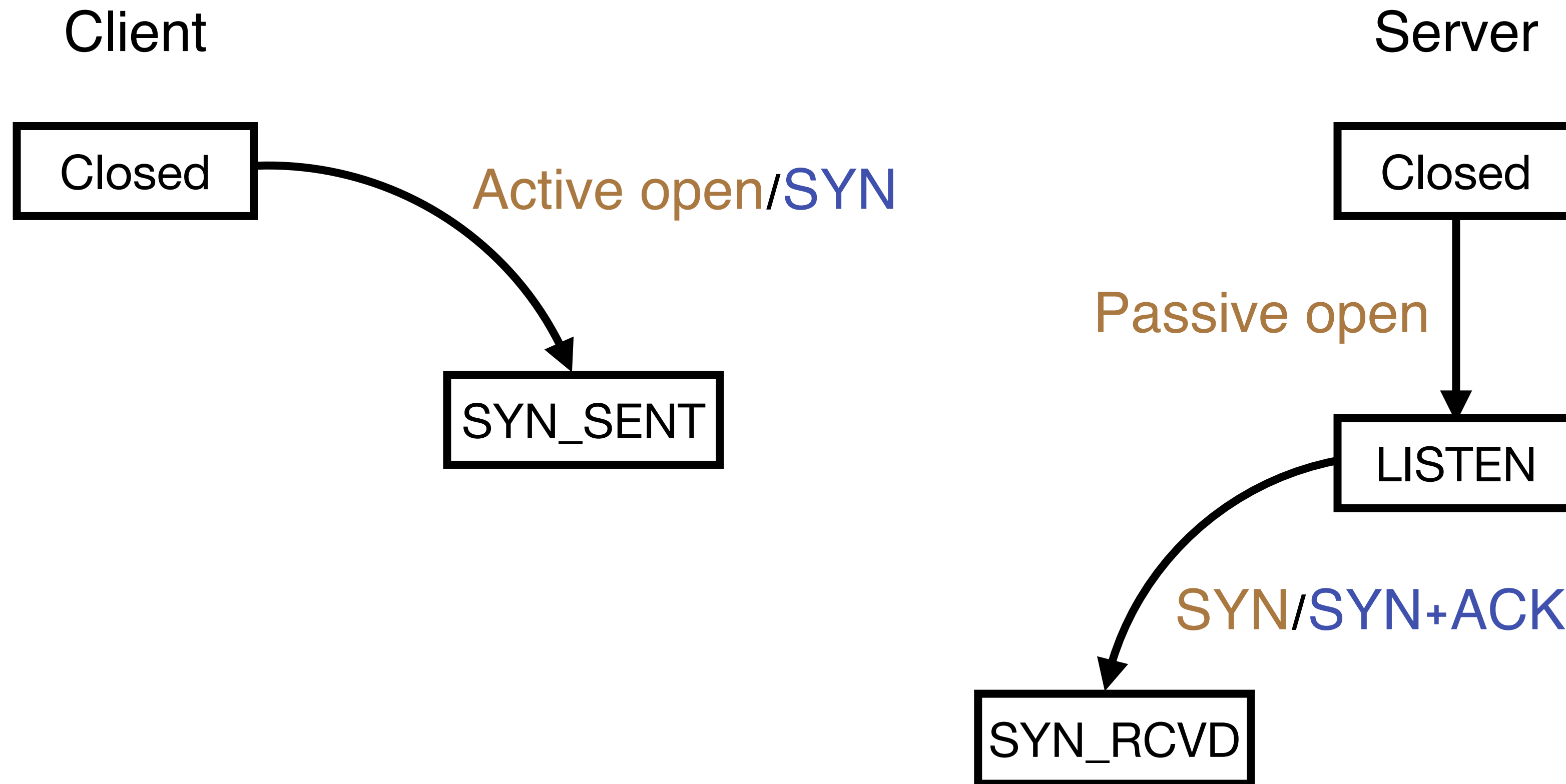
# State Machine (event/action)



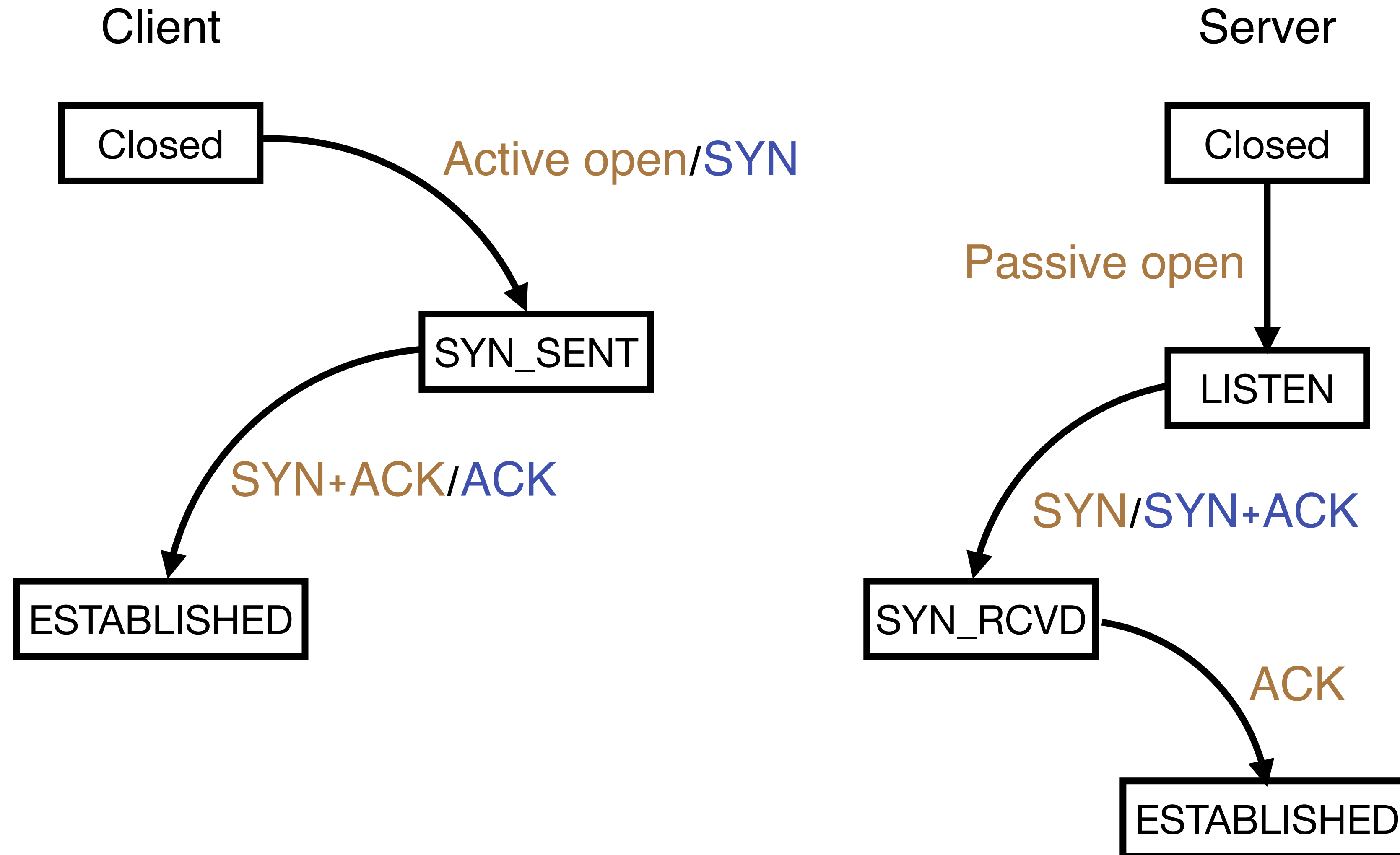
# State Machine (Step 1)



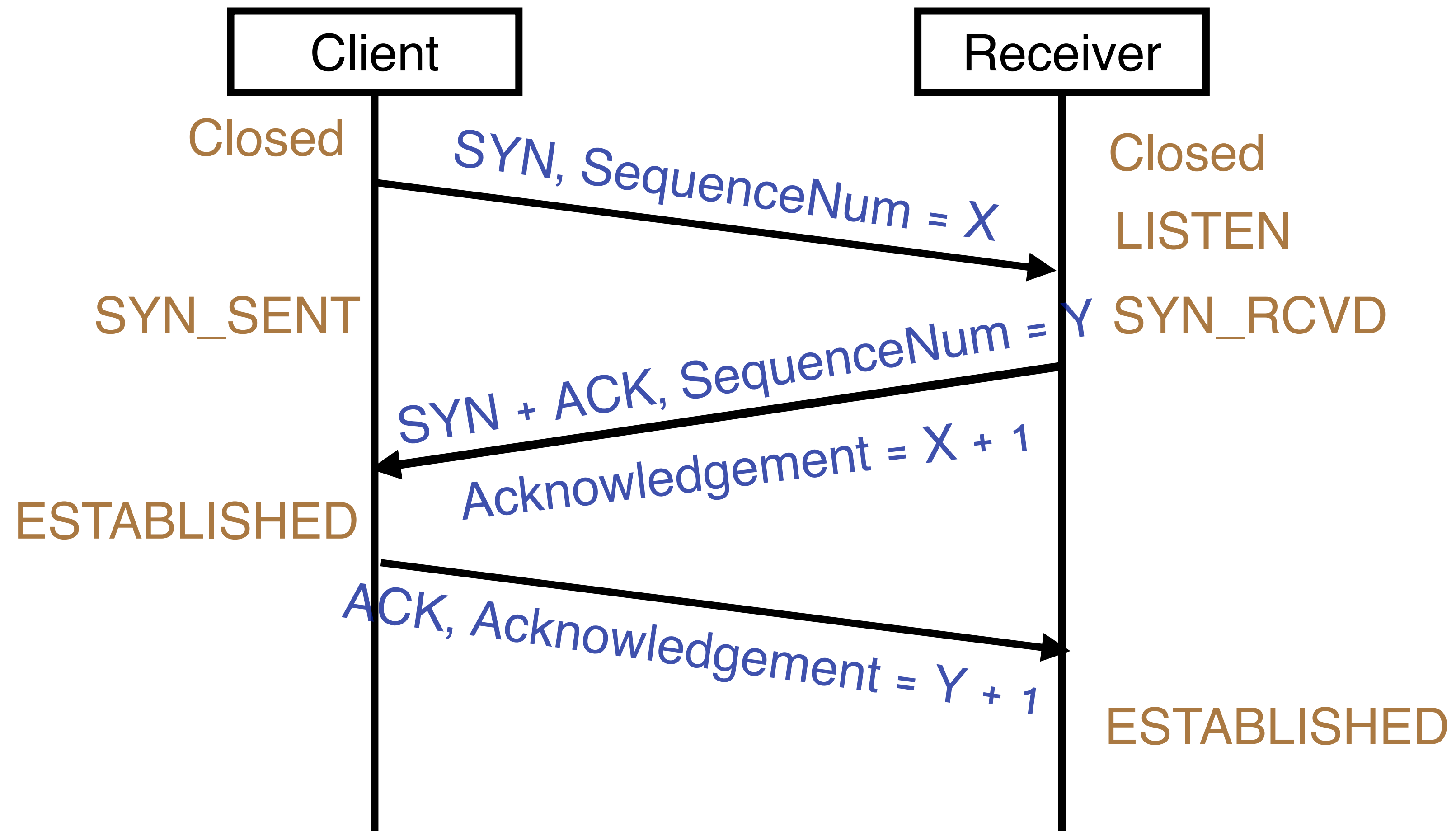
# State Machine (Step 2)



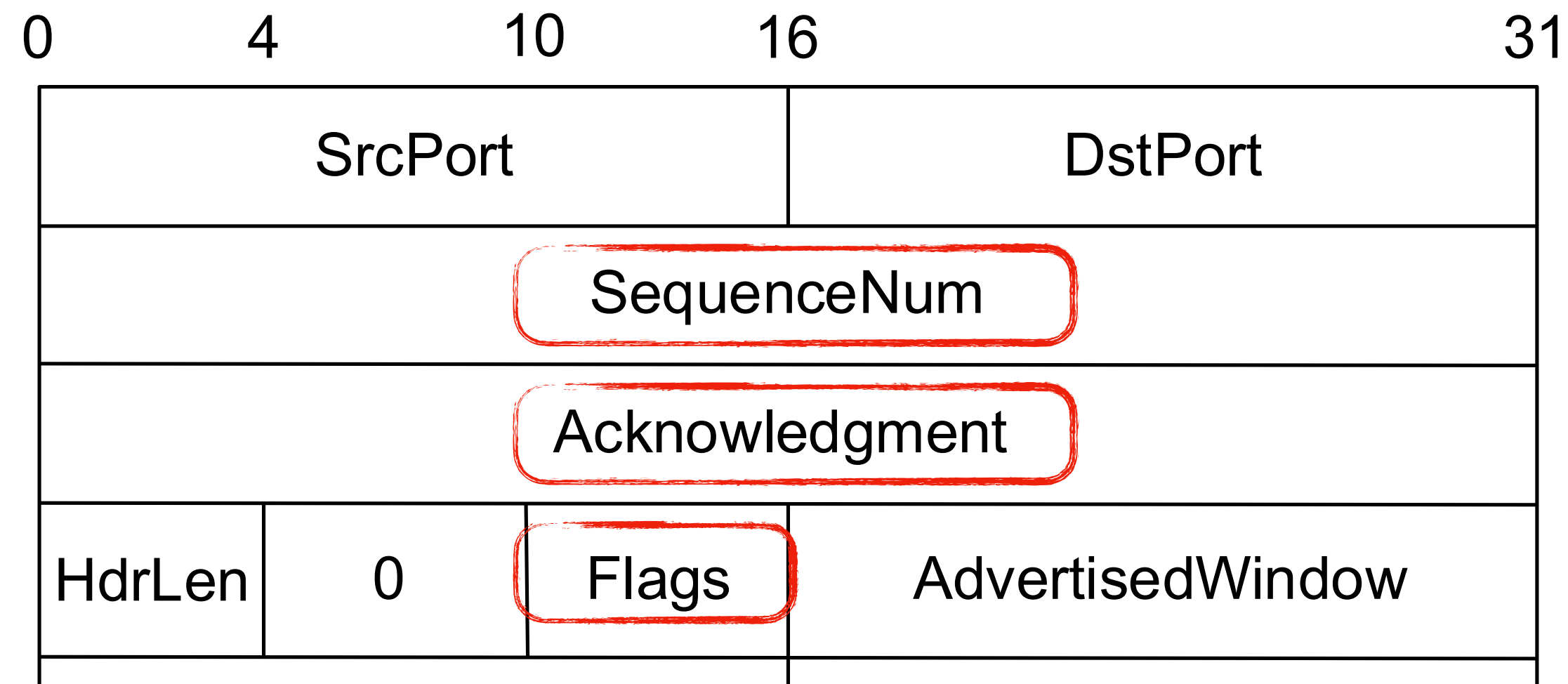
# State Machine (Step 3)



# TCP Connection Establishment Summary

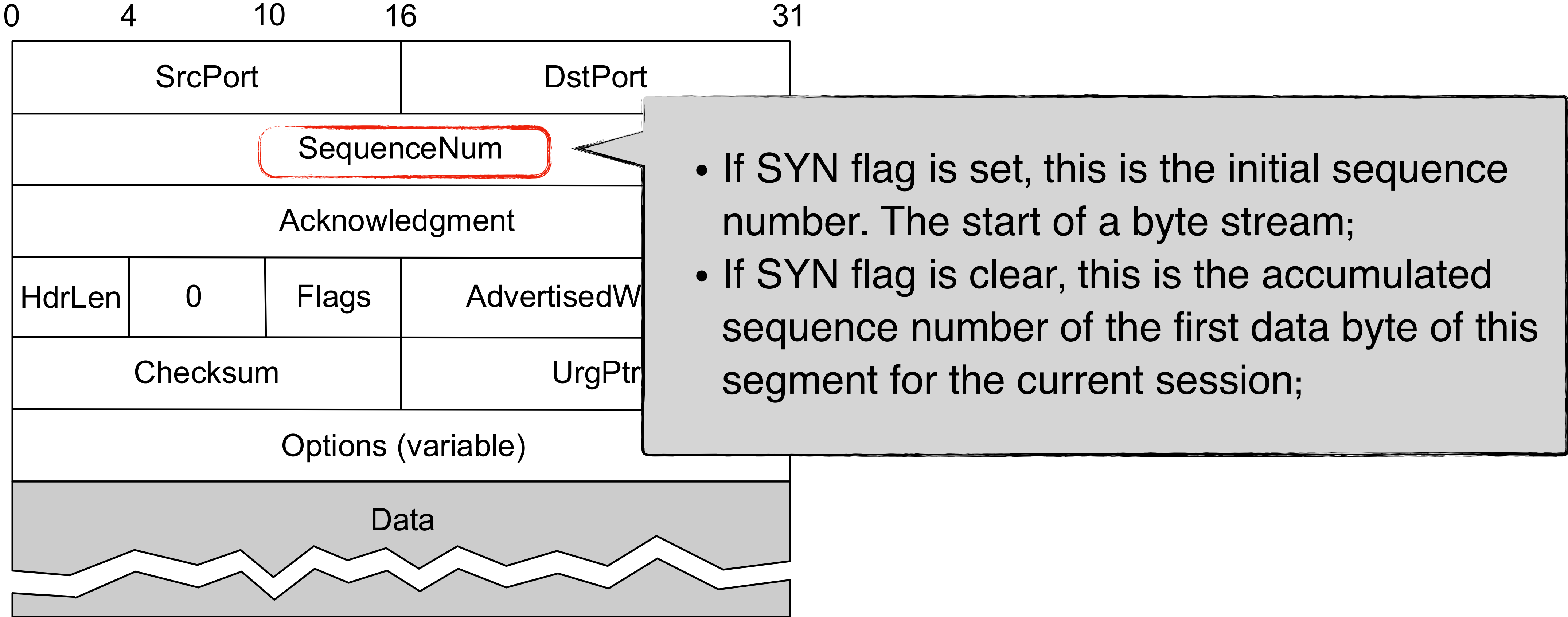


# Revisit the TCP Header

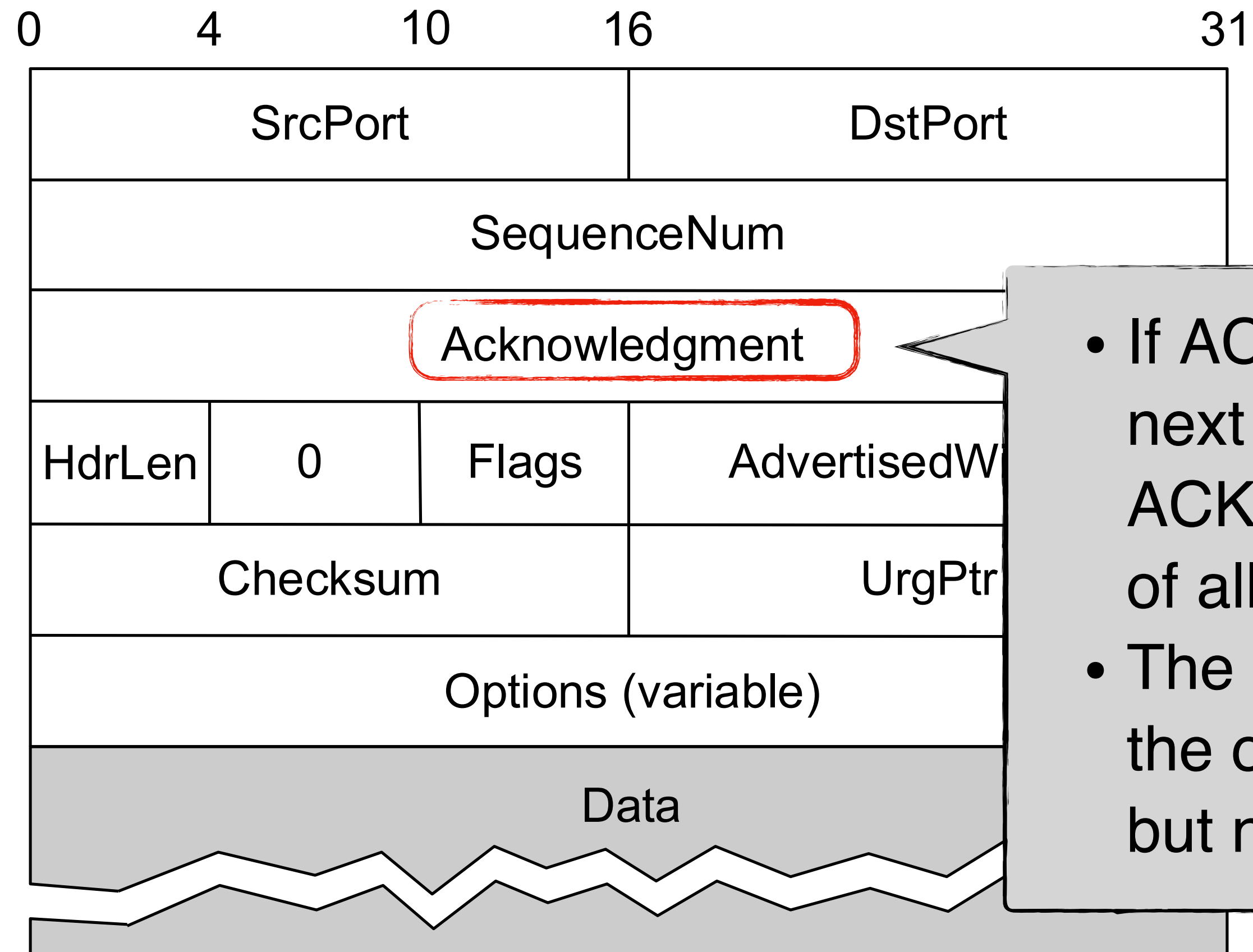


- SYN/FIN -> TCP connection establishment and teardown
- ACK -> Acknowledgement is valid
- URG -> The segment contains urgent data. UrgPtr will be setup
- PUSH -> Notify the receiving process
- RESET -> The receiving side gets confused information

# Revisit the TCP header



# Revisit the TCP header



- If ACK flag is set, the value of this field is the next sequence number that the sender of the ACK is expecting. This acknowledges receipt of all prior bytes (if any)
- The first ACK sent by each end acknowledges the other ends's initial sequence number itself, but no data



## 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

## 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. Border gateway protocol (BGP)
25. Network address translation (NAT)
26. User Datagram Protocol (UDP)
27. Transmission Control Protocol (TCP)
28. Three-way Handshake

# Summary

## Today's takeaways

- #1: TCP connection setup uses a three-way handshake to build a duplex channel
- #2: The three-way handshake can be implemented via a state machine

## Next lecture

- TCP connection teardown