ICD Course SUT

Instructor: Dr. Foshati

TCP PROTOCOL

Amirkasra Ahmadi

What is TCP

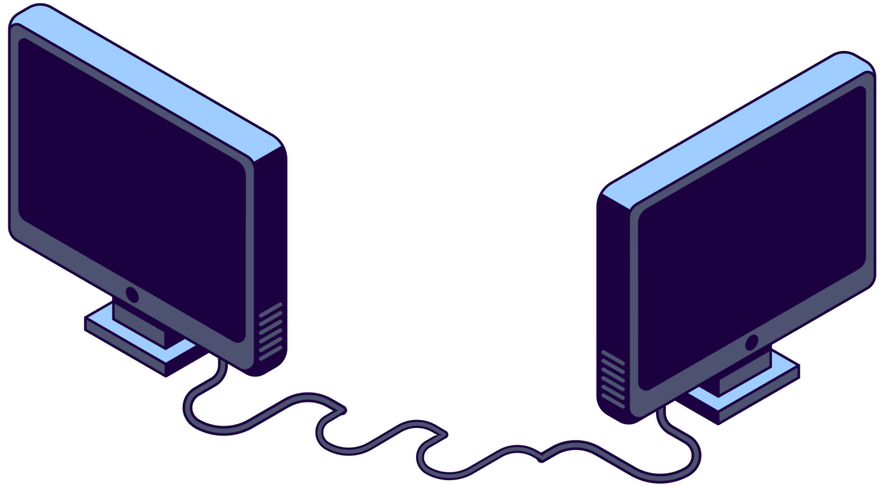
Stands for Transmission Control Protocol

One of the core Internet protocols (part of TCP/IP suite)

Ensures reliable, ordered, and error-checked data delivery between applications

Why was TCP Developed?

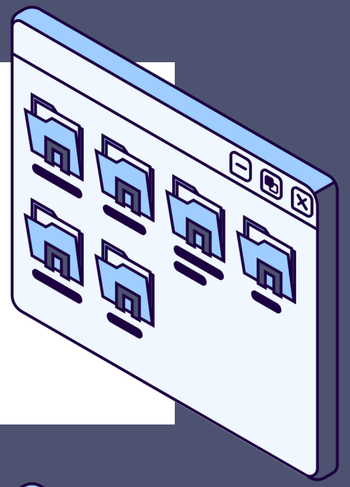
TCP was developed in the early 1970s by Vinton Cerf and Robert Kahn to address the need for reliable communication across large-scale and heterogeneous networks, such as the Internet.



It was designed to ensure reliable data transmission by incorporating mechanisms for error correction, congestion control, and flow control, making it a fundamental protocol for modern network communication.

TCP Characteristics

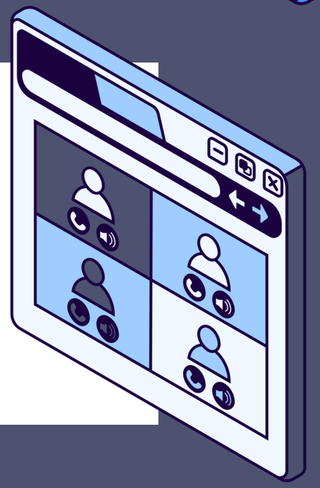
Connection-Oriented: Establishes a connection before transmitting data



Reliable: Guarantees that data arrives intact and in order

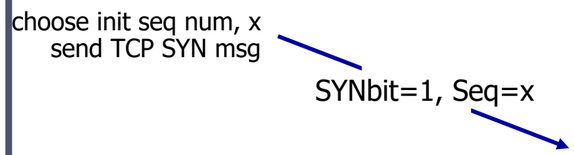
Full-Duplex: Allows simultaneous data transfer in both directions

Stream-Based: Transfers a continuous stream of bytes rather than fixed-sized packets



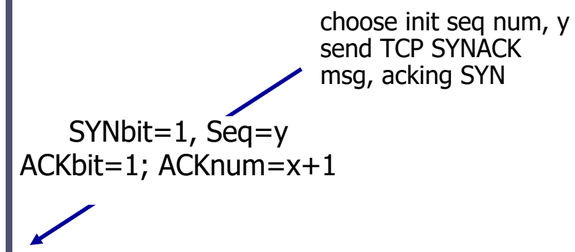
TCP Three-Way Handshake

SYN



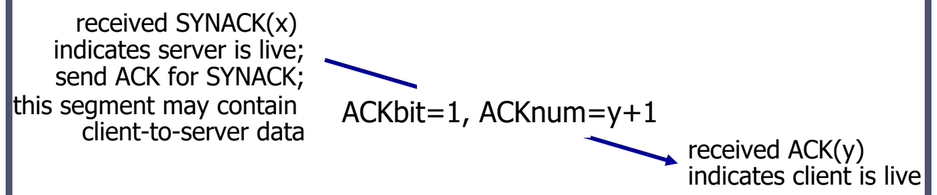
Request Initiate Connection

SYN-ACK



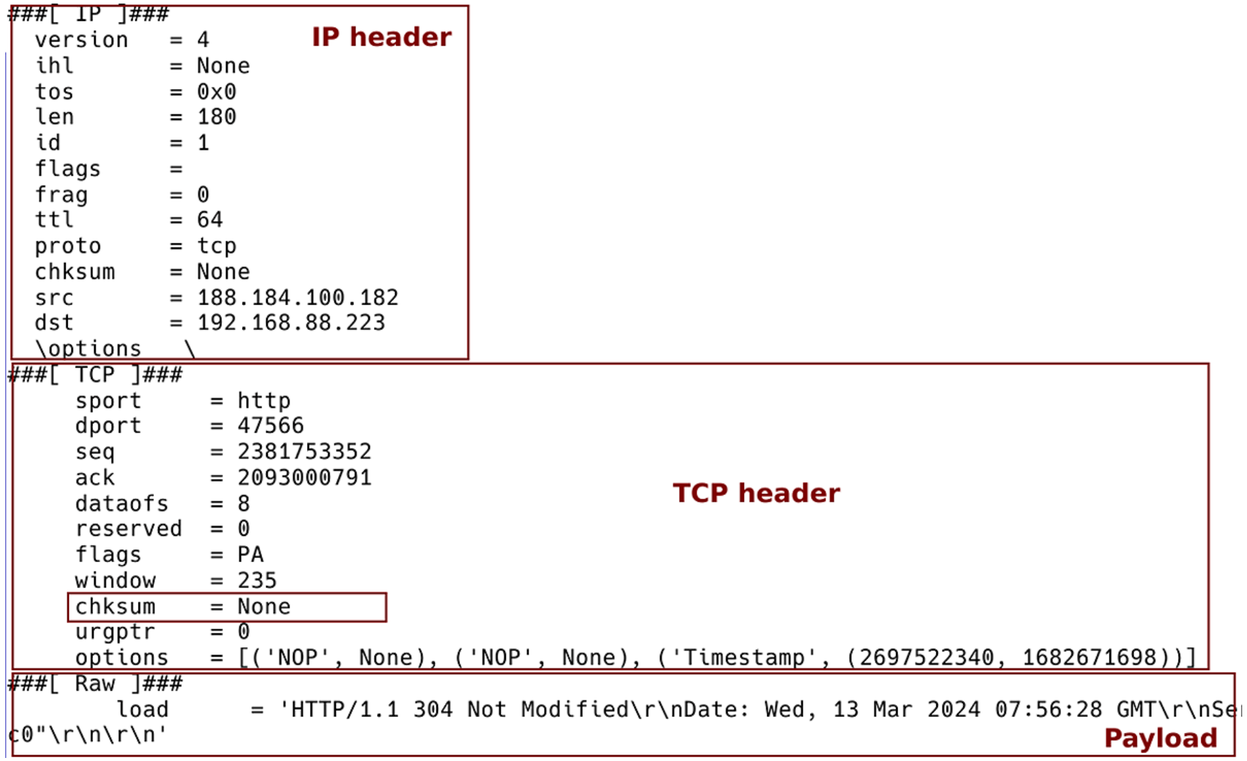
Respond to SYN

ACK



Sent By Client to say connection is established.

Error DETECTION IN TCP

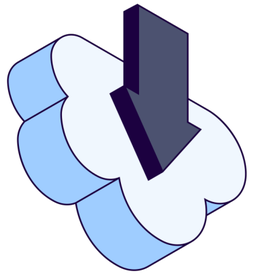


Sender calculates and includes the checksum in the segment.

Receiver recalculates the checksum and verifies integrity.

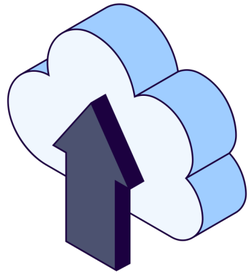
Error Correction in TCP ❌

Drops



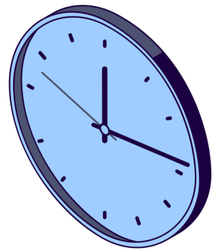
Drops the corrupted segment

Retransmission

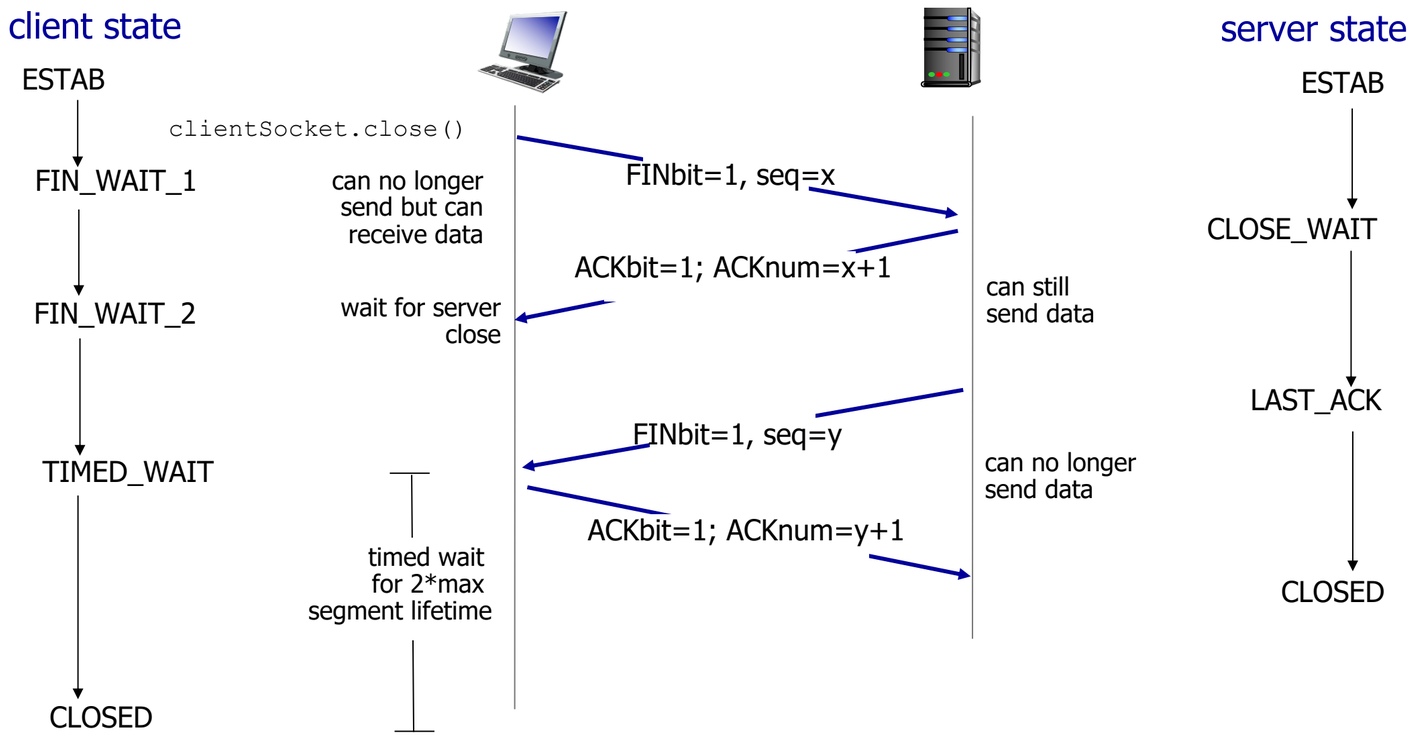


Requests retransmission

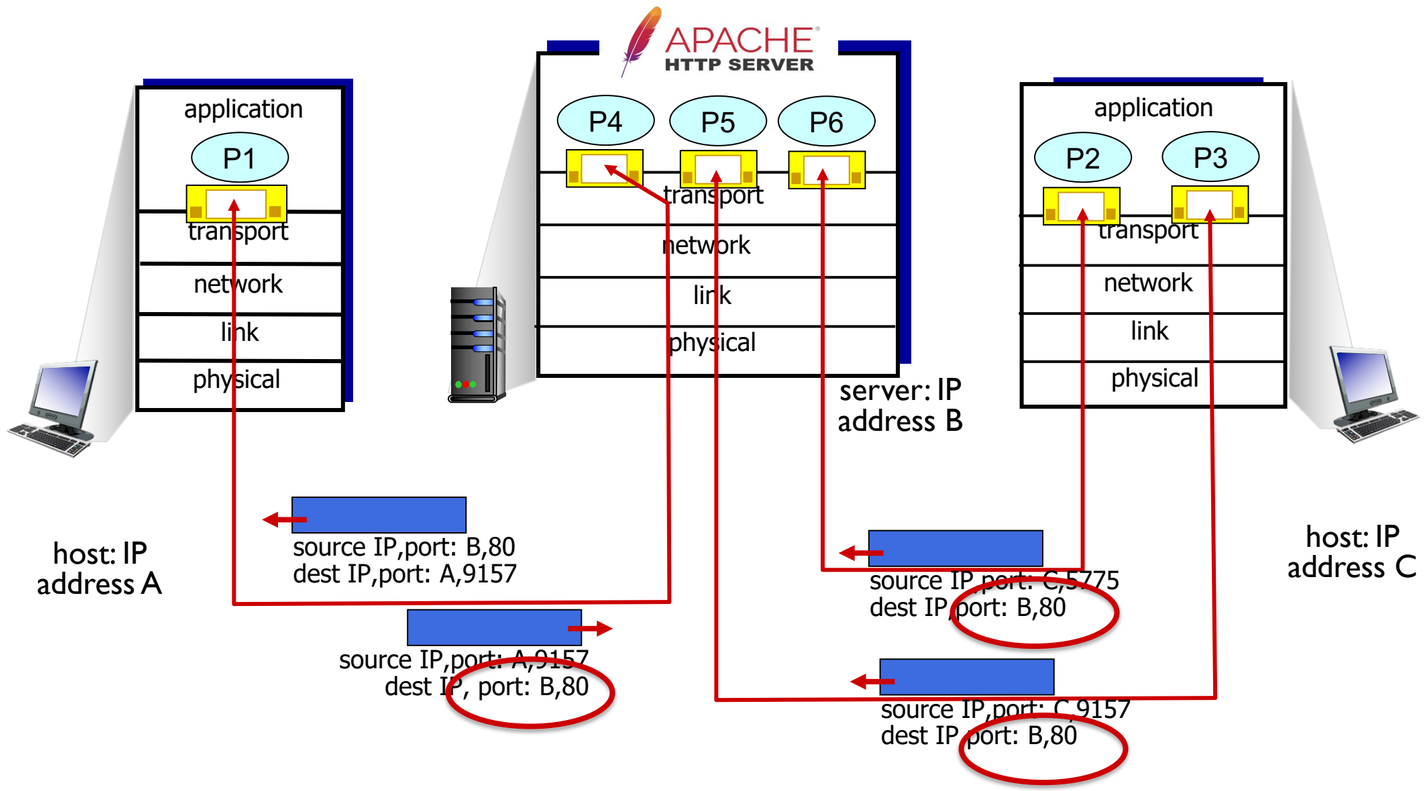
Acknowledgment



Uses acknowledgment numbers to confirm received data



Closing a TCP connection



Port numbers identify specific applications or services on a device

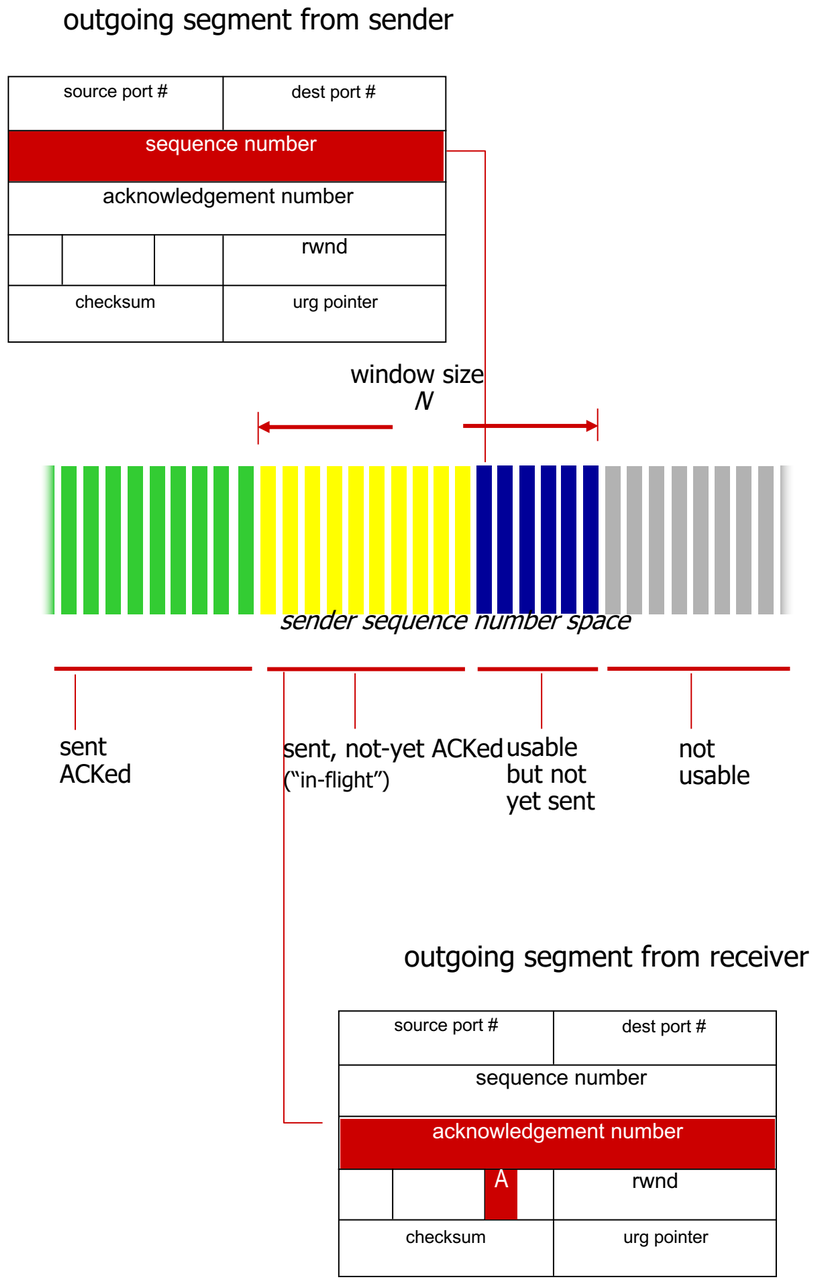
Each TCP connection uses:

- Source Port and Destination Port.
- Example: HTTP uses port 80, FTP uses port 21.

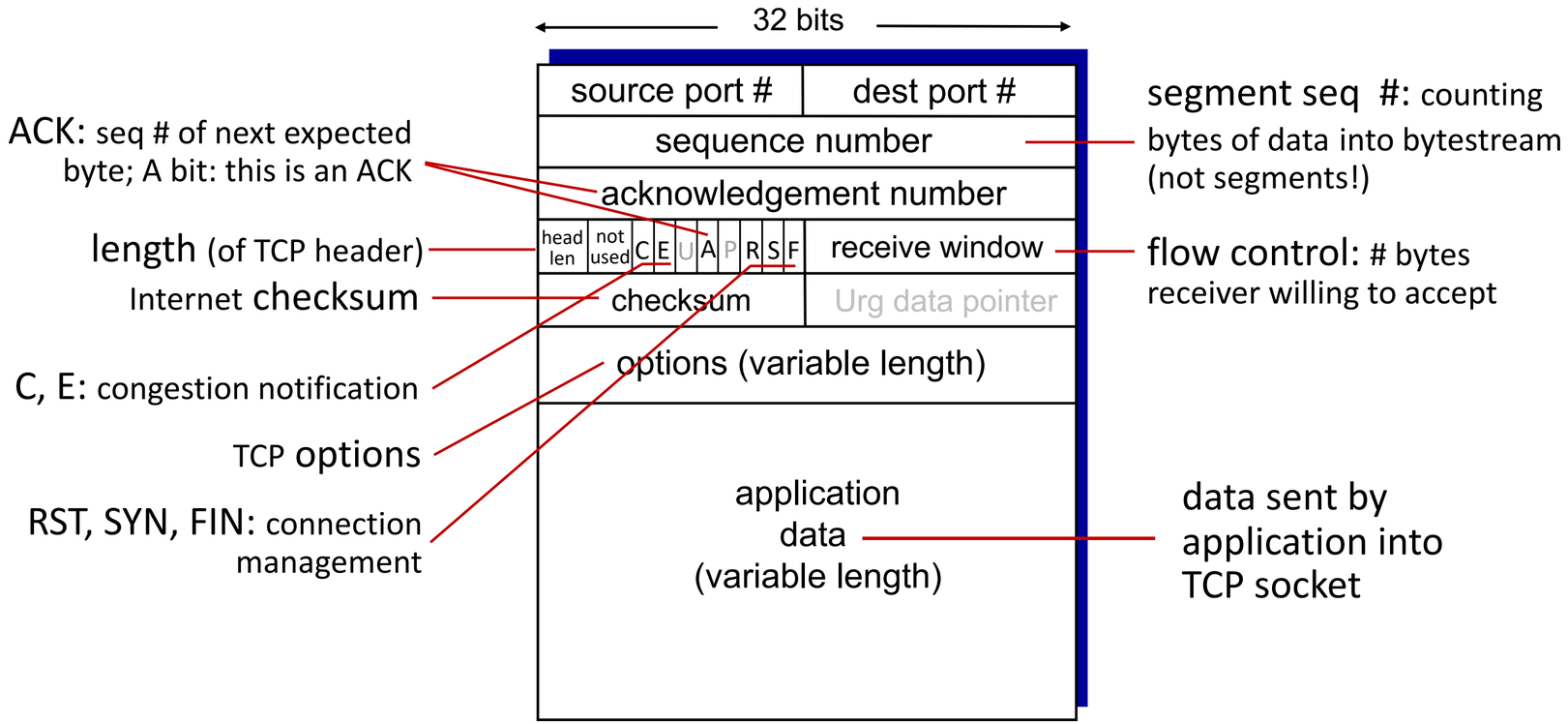
Role of Port Numbers in TCP

How Does TCP Handle Data Transmission in a Network

Sequence Number	**Each segment has a unique sequence number**
ACK	**The receiver sends back an ACK for each segment.**



TCP Segment Structure

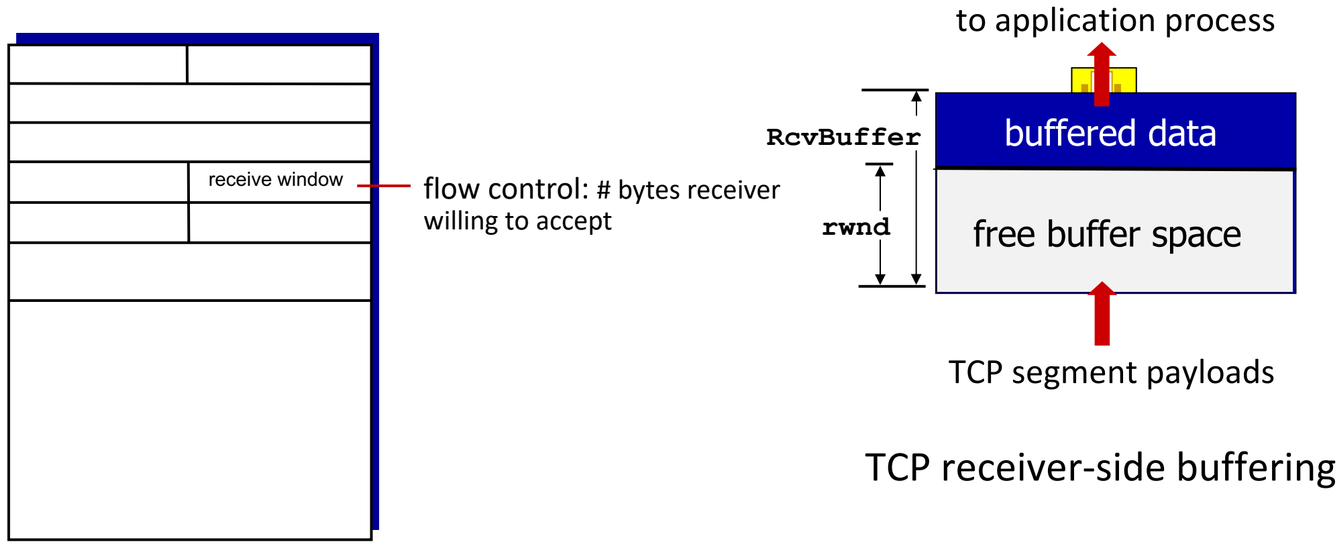


Flow Control in TCP

Uses a Sliding Window Mechanism

Receiver informs sender of available buffer size (Window Size field)

Prevents buffer overflow at the receiver

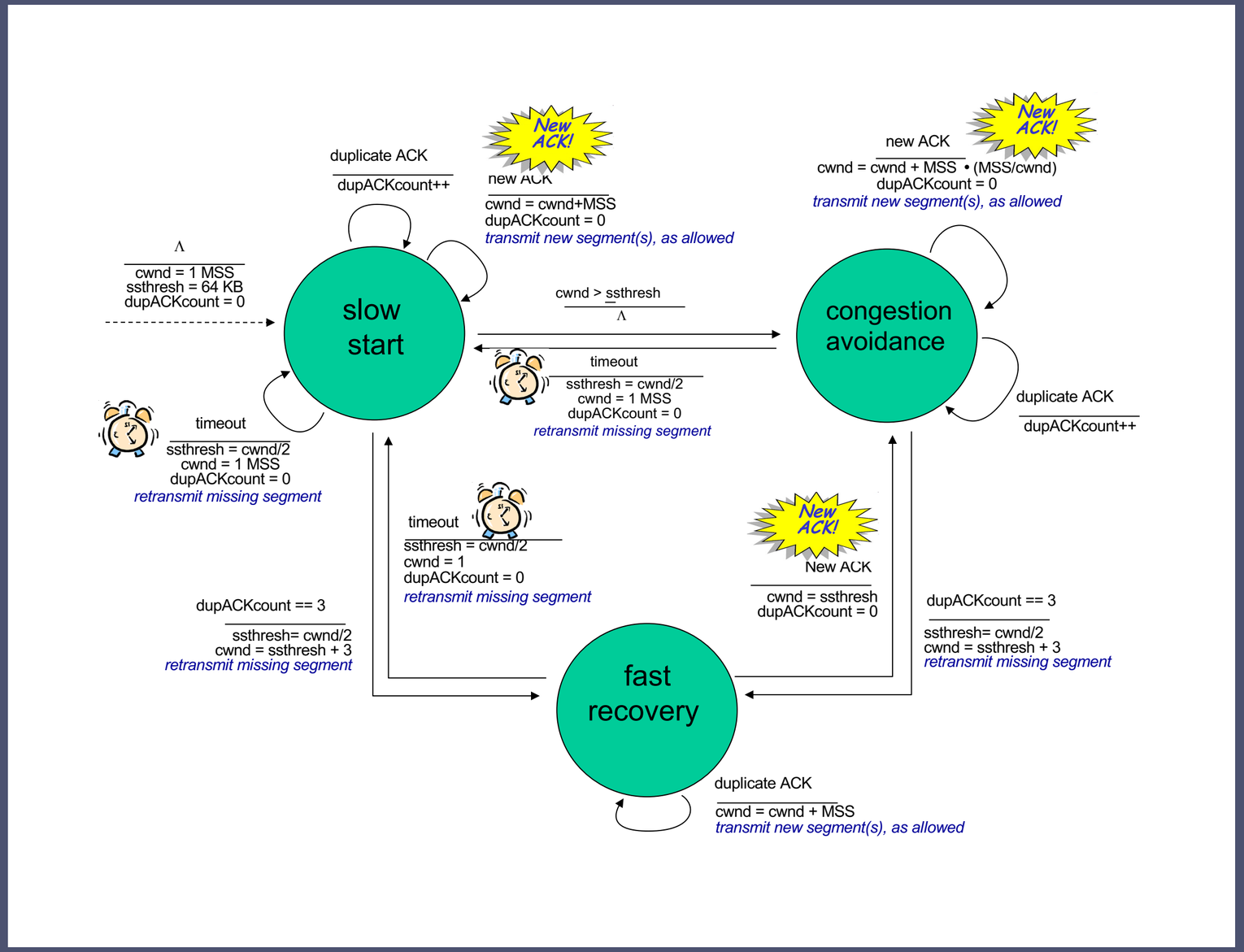


TCP receiver-side buffering

Congestion Avoidance in TCP

Slow Start	Congestion Avoidance	Fast Recovery
Gradually increases transmission rate	Reduces rate when congestion is detected	Quickly recovers from packet loss

TCP Congestion Control



TCP Congestion Control

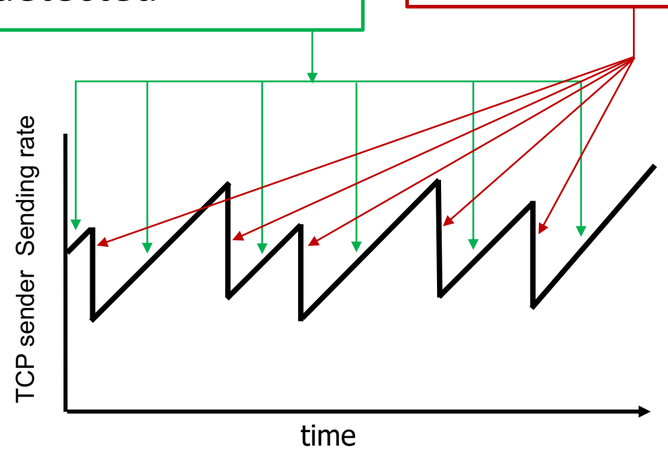
TCP CUBIC

Additive Increase

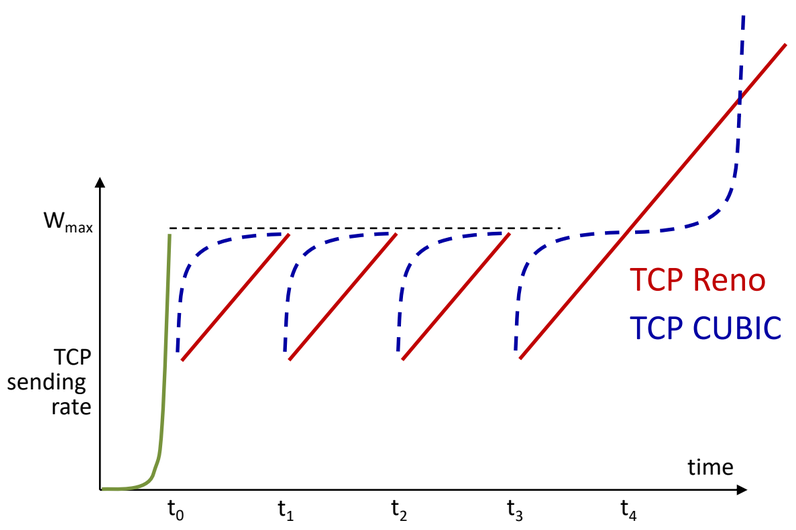
increase sending rate by 1 maximum segment size every RTT until loss detected

Multiplicative Decrease

cut sending rate in half at each loss event



TCP CUBIC



THANKS FOR YOUR ATTENTION