

# TCP PROTOCOL

# What is TCP

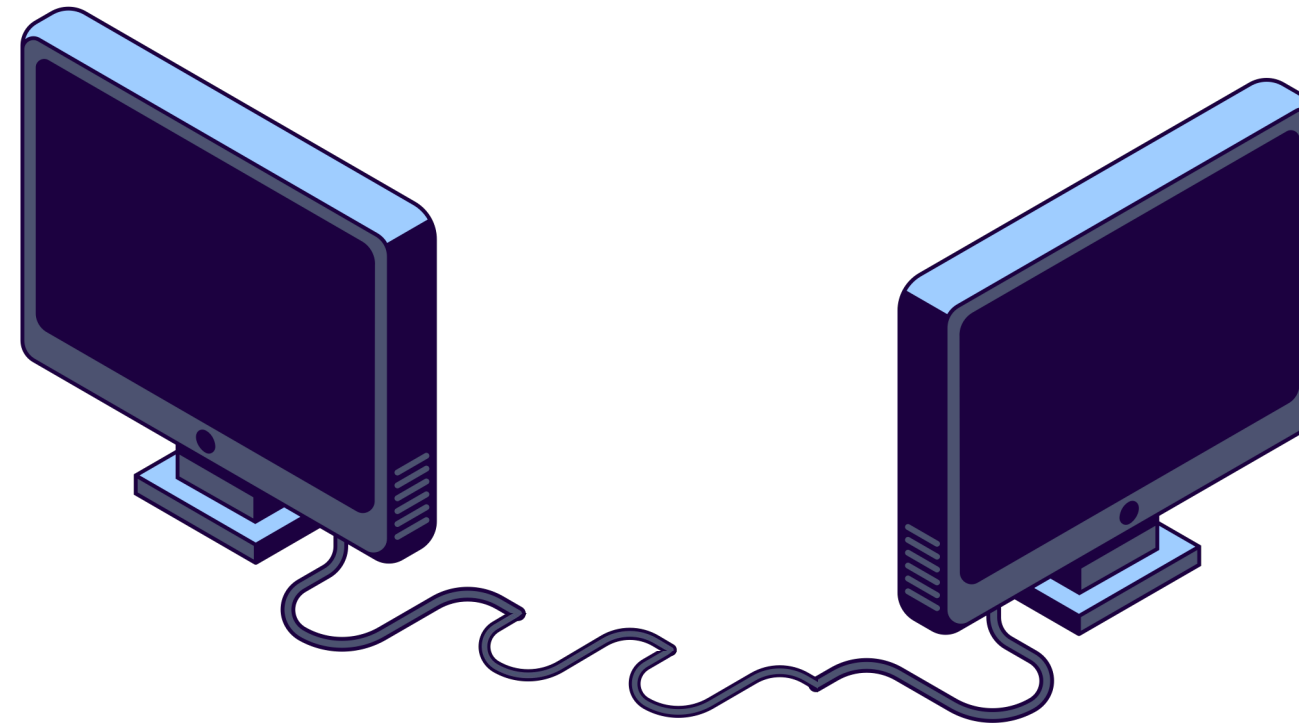
Stands for Transmission Control Protocol

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?

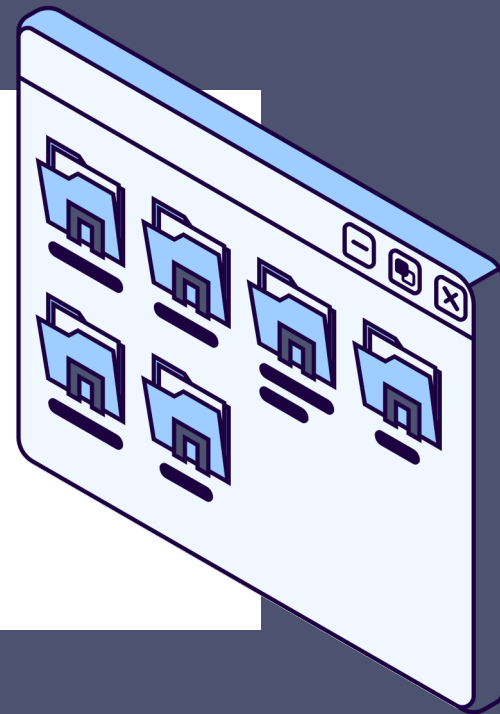
TCP was developed in the early 1970s by Vinton Cerf and Robert Kahn to address the need for reliable communication across large-scale and heterogeneous networks, such as the Internet.



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



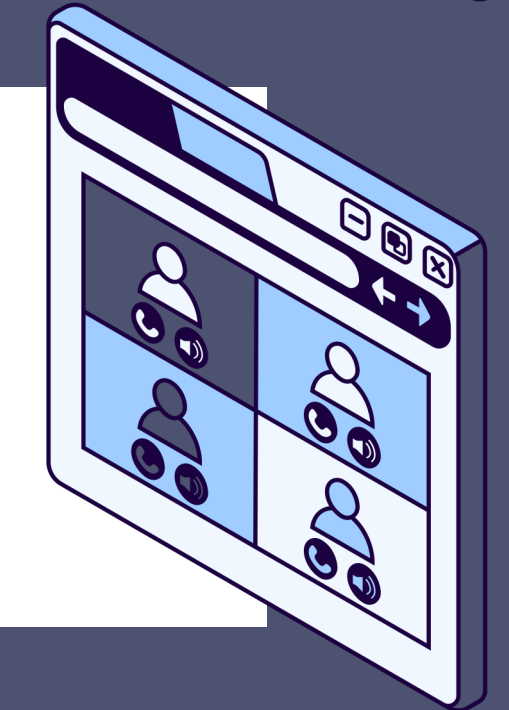
Reliable: Guarantees that data arrives intact and in order



Full-Duplex: Allows simultaneous data transfer in both directions



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



## SYN

choose init seq num, x  
send TCP SYN msg

SYNbit=1, Seq=x

Request Initiate  
Connection

## SYN-ACK

choose init seq num, y  
send TCP SYNACK  
msg, acking SYN

SYNbit=1, Seq=y  
ACKbit=1; ACKnum=x+1

Respond to SYN

## ACK

received SYNACK(x)  
indicates server is live;  
send ACK for SYNACK;  
this segment may contain  
client-to-server data

ACKbit=1, ACKnum=y+1

received ACK(y)  
indicates client is live

Sent By Client to  
say connection is  
established.

# TCP Three-Way Handshake

# Error DETECTION IN TCP

```
###[ IP ]###
version  = 4          IP header
ihl      = None
tos      = 0x0
len      = 180
id       = 1
flags    =
frag     = 0
ttl      = 64
proto    = tcp
chksum   = None
src      = 188.184.100.182
dst      = 192.168.88.223
\options \

###[ TCP ]###
sport    = http
dport    = 47566
seq      = 2381753352
ack      = 2093000791
dataofs  = 8          TCP header
reserved = 0
flags    = PA
window   = 235
chksum   = None
urgptr   = 0
options  = [('NOP', None), ('NOP', None), ('Timestamp', (2697522340, 1682671698))]

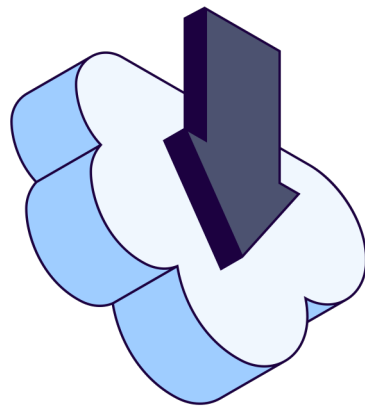
###[ Raw ]###
load     = 'HTTP/1.1 304 Not Modified\r\nDate: Wed, 13 Mar 2024 07:56:28 GMT\r\nSe
c0"\r\n\r\n'          Payload
```

Sender calculates and includes the checksum in the segment.

Receiver recalculates the checksum and verifies integrity.

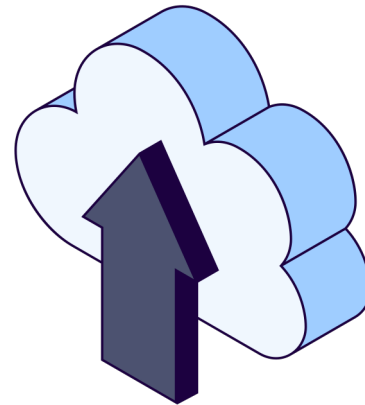
# Error Correction in TCP

## Drops



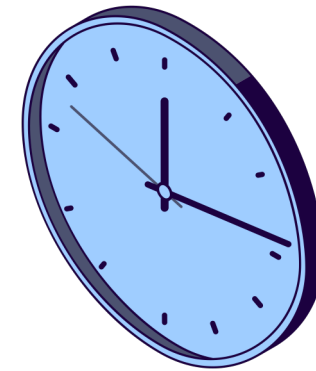
Drops the corrupted  
segment

## Retransmission

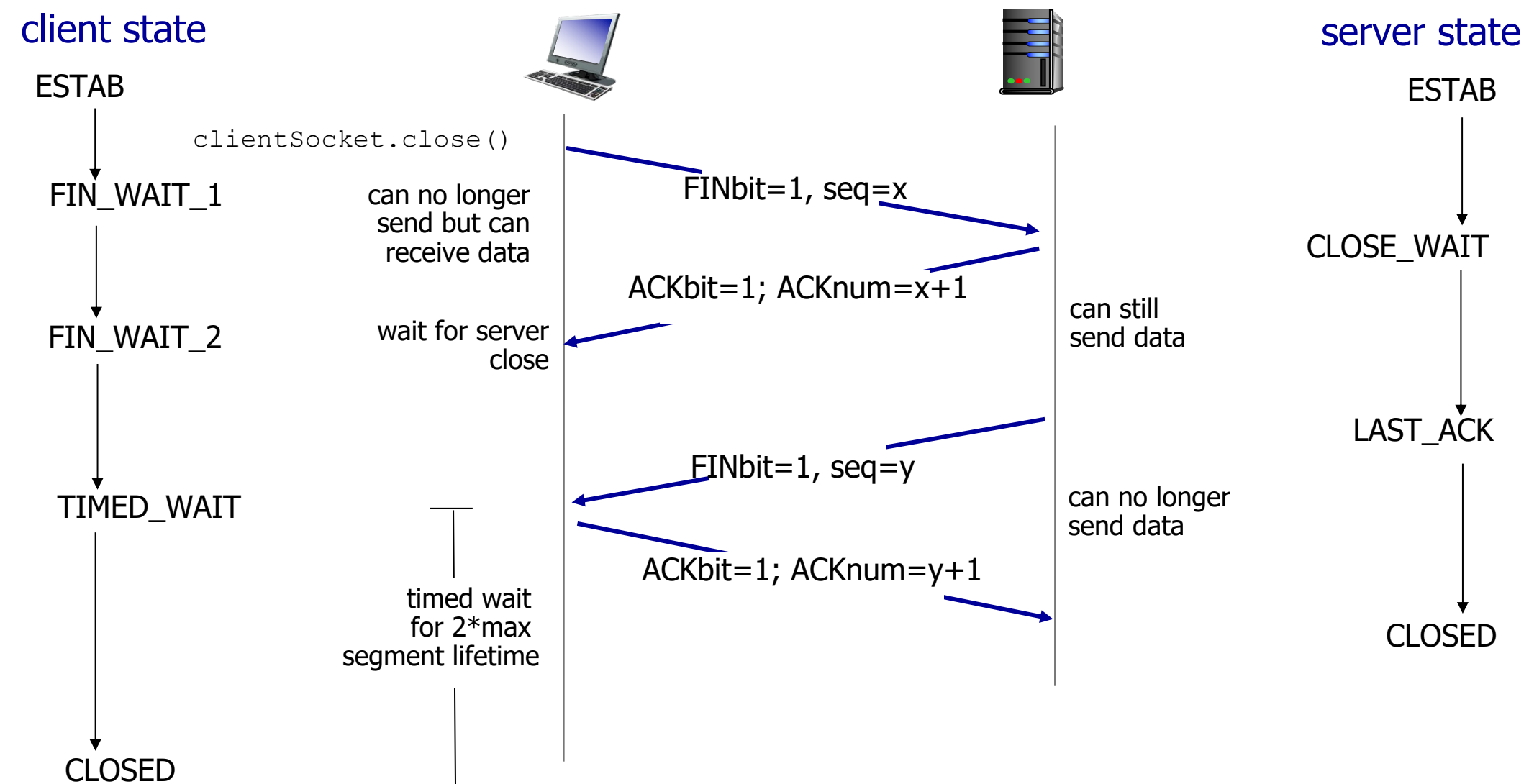


Requests  
retransmission

## Acknowledgment

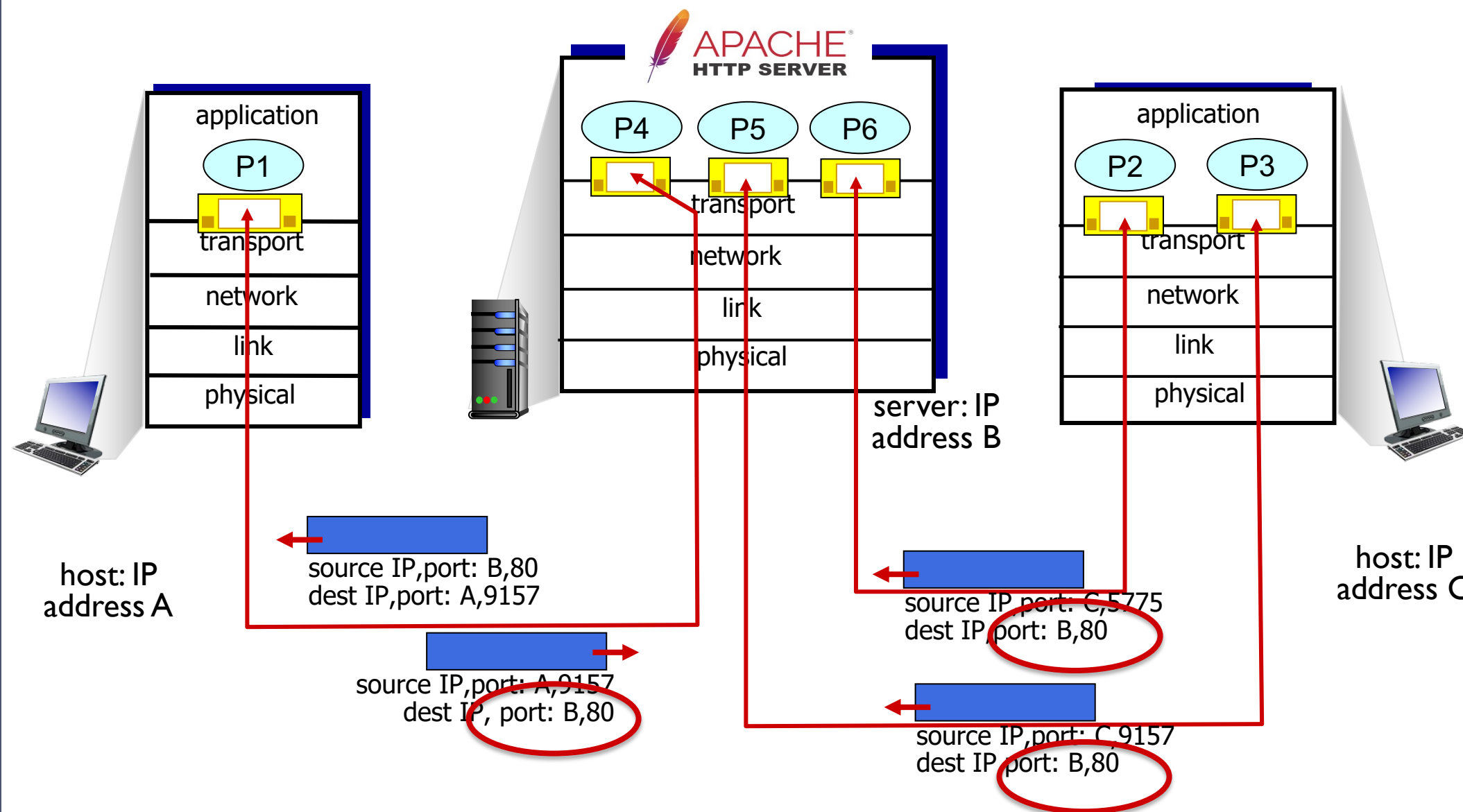


Uses  
acknowledgment  
numbers to confirm  
received data



# 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

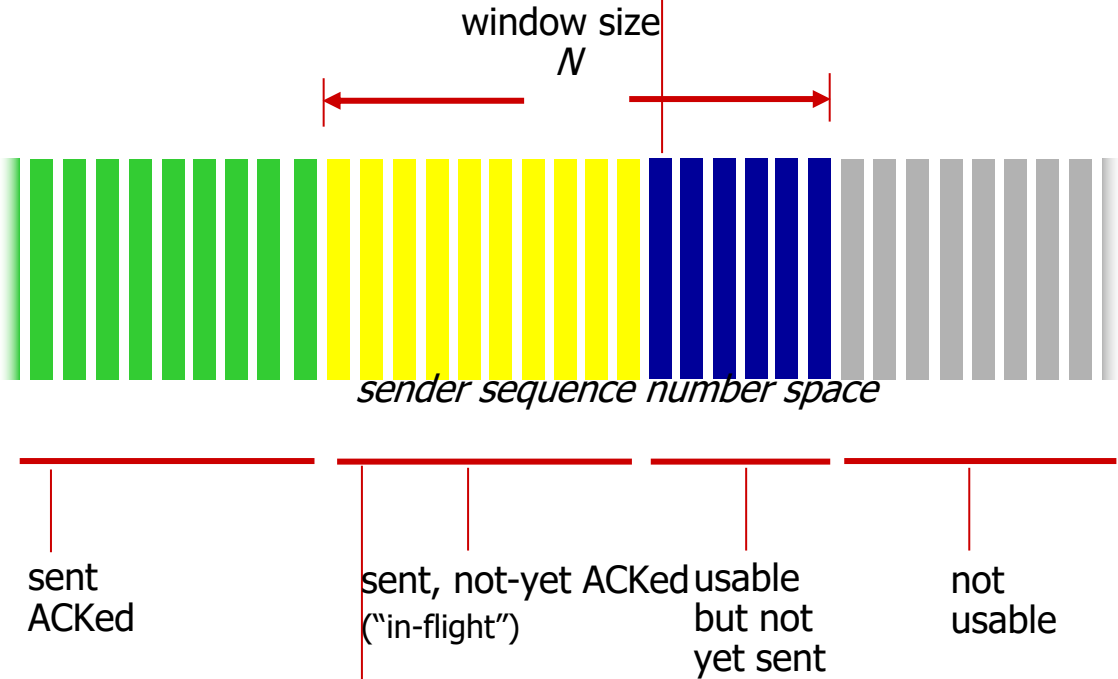
Each segment has a unique sequence number

## ACK

The receiver sends back an ACK for each segment.

outgoing segment from sender

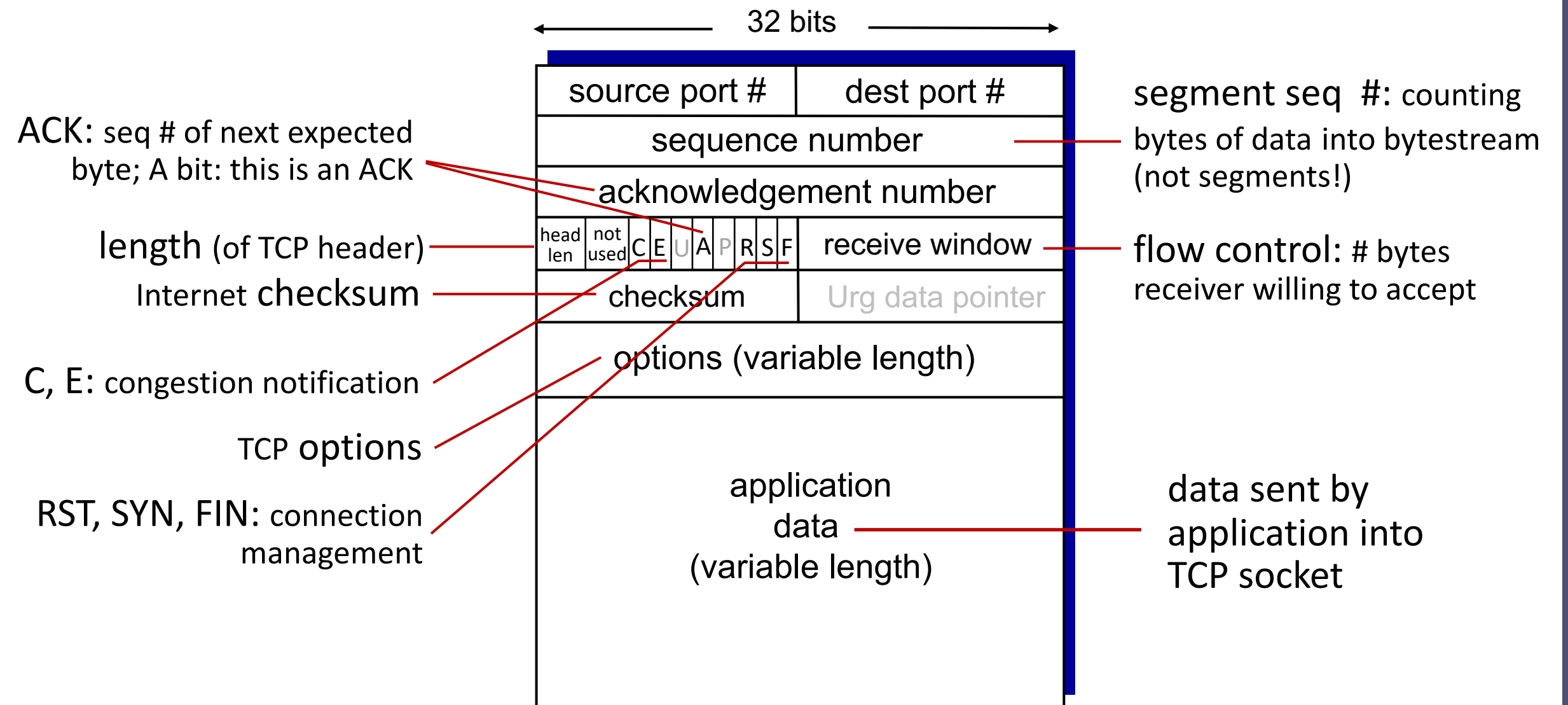
source port #		dest port #	
sequence number			
acknowledgement number			
		rwnd	
checksum		urg pointer	



outgoing segment from receiver

source port #		dest port #	
sequence number			
acknowledgement number			
		A	rwnd
checksum		urg pointer	

# TCP Segment Structure

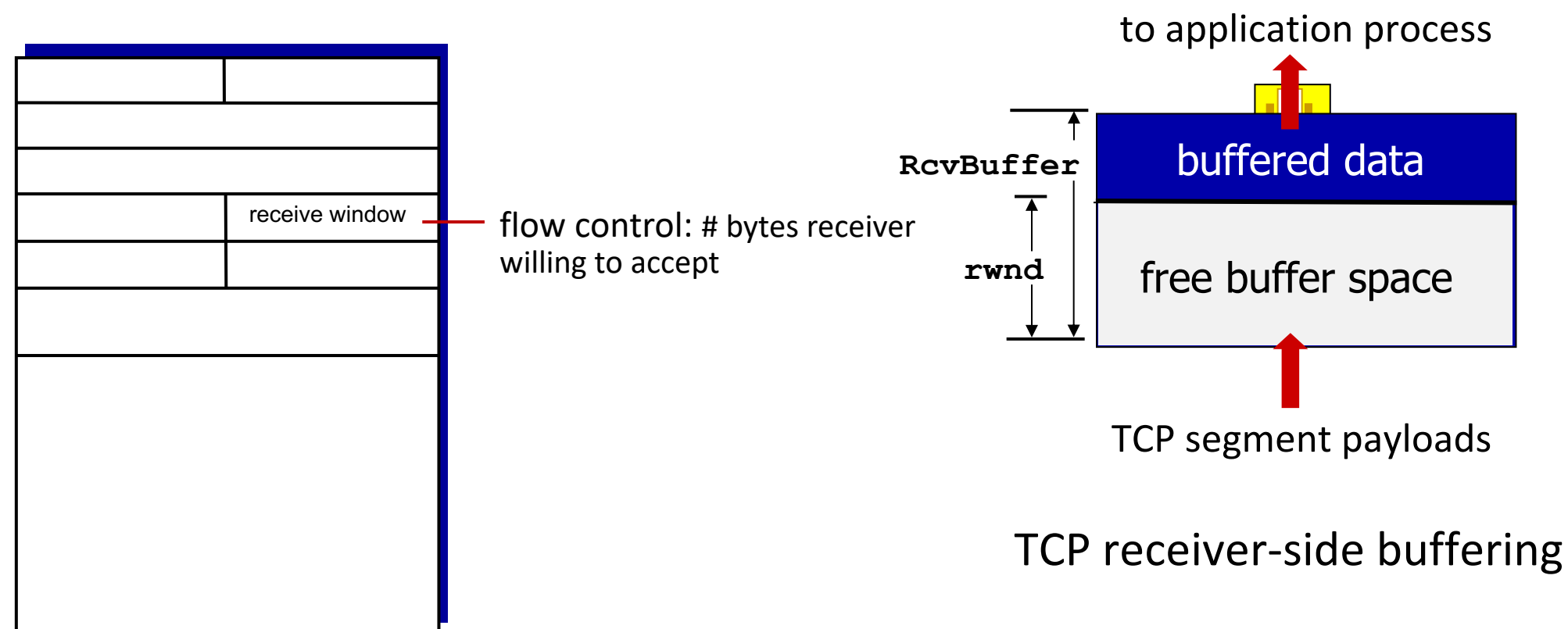


Uses a Sliding Window Mechanism

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

Prevents buffer overflow at the receiver

# Flow Control in TCP



# Congestion Avoidance in TCP

## Slow Start

Gradually increases  
transmission rate

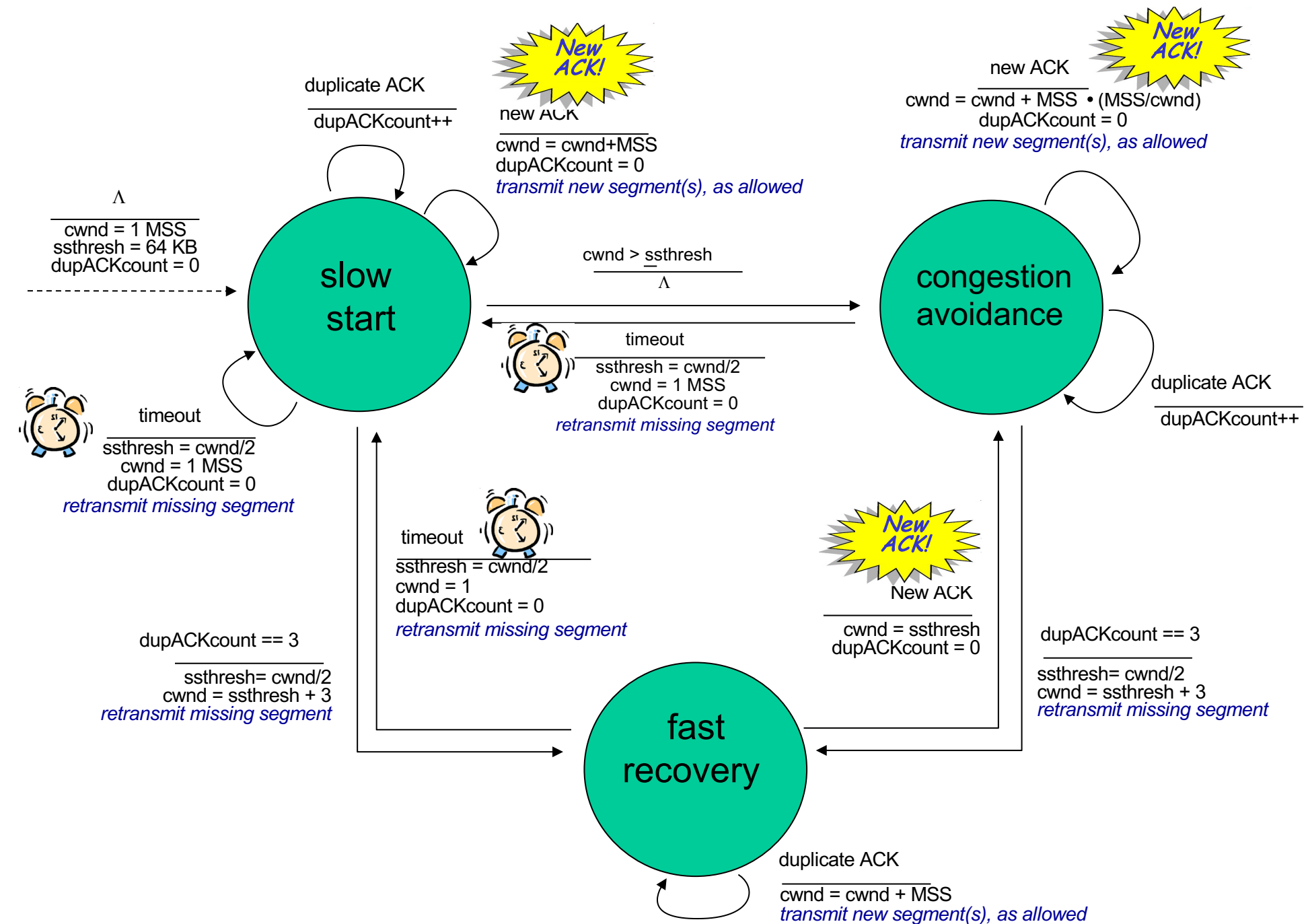
## Congestion Avoidance

Reduces rate when  
congestion is  
detected

## Fast Recovery

Quickly recovers  
from packet loss

# TCP Congestion Control



# TCP Congestion Control

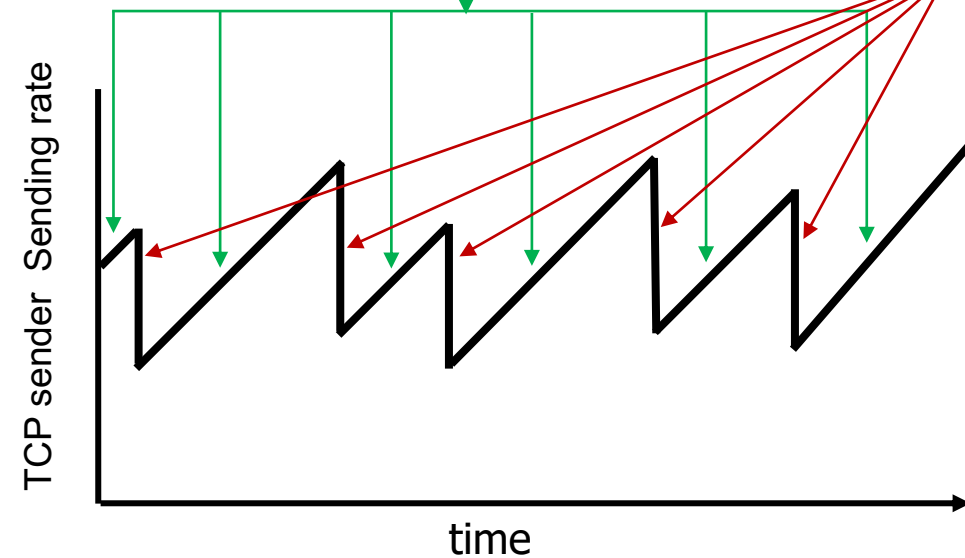
## TCP CUBIC

### Additive Increase

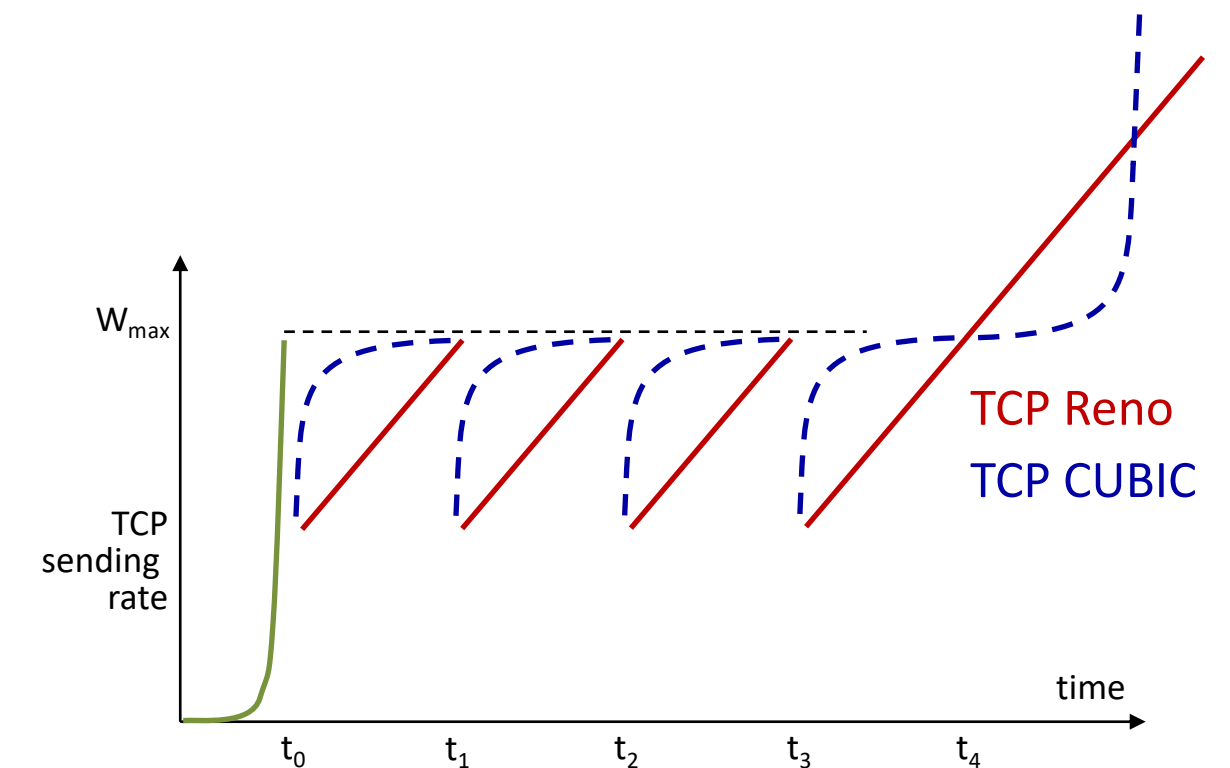
increase sending rate by 1  
maximum segment size every  
RTT until loss detected

### Multiplicative Decrease

cut sending rate in half at  
each loss event



## TCP CUBIC



**THANKS FOR  
YOUR  
ATTENTION**