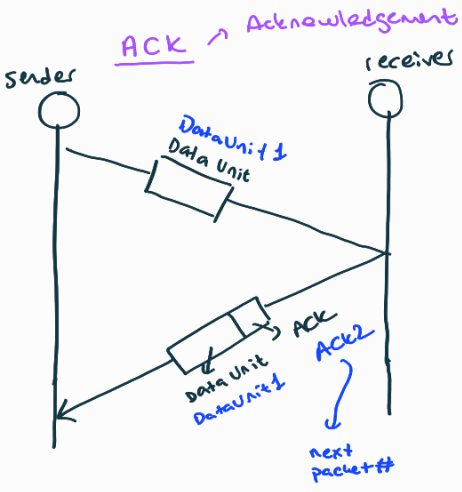
TCP - Reliable Transmission

References: ECE Güran Schmidt's EE444 Lecture Slides



Ideal Network

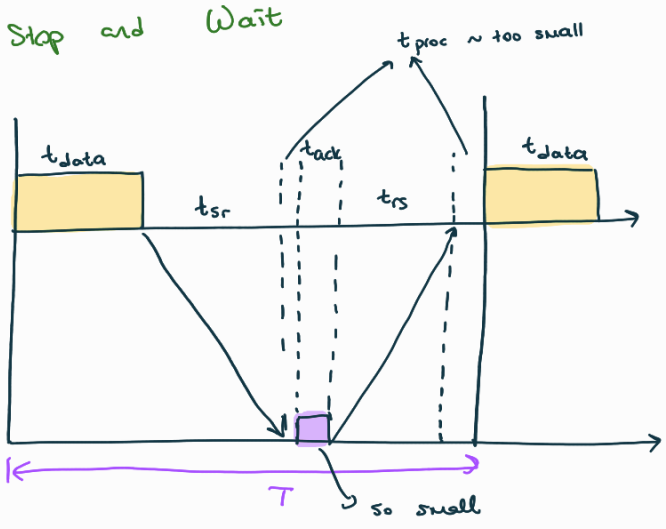
↓

- Error free transmission
- Infinite buffer
- No acknowledgement is necessary
- No flow control

- If sender receives the ACK's → goes on transmitting
- If sender fails to get the ACKs → decides sent is wrong, retransmit

Some Flow Control Algorithms

1) Stop and Wait



$$\text{Efficiency}(\eta) = \frac{t_{data}}{(t_{data} + t_{sr} + t_{rs})}$$

↓

Fraction of time that is spent directly for the user data

Automatic Repeat Request (ARQ) Error Control

→ Individual or Piggybacked Cumulative ACKs

→ Pipelining (sender $w > 1$, receiver $w > 1$)

→ Sequence Numbering

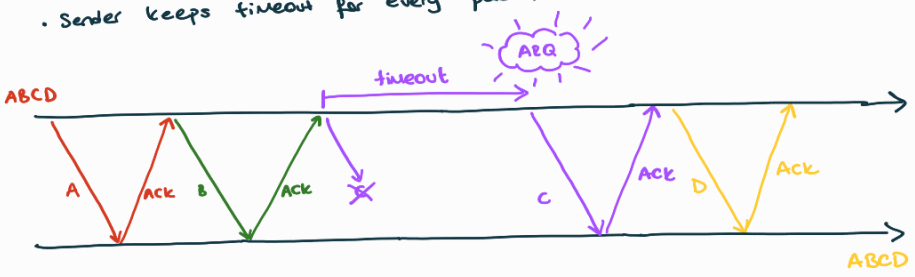
→ Retransmission triggered by:

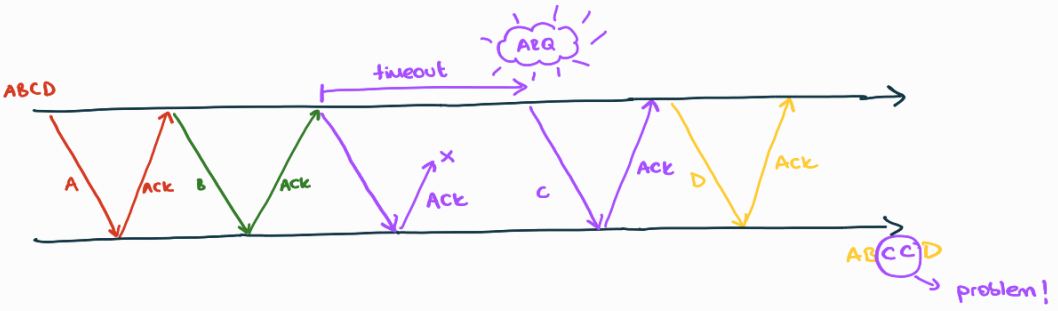
- Time-out
- Duplicate ACK
- NACK

1.1) Stop and Wait for transmission with errors and losses

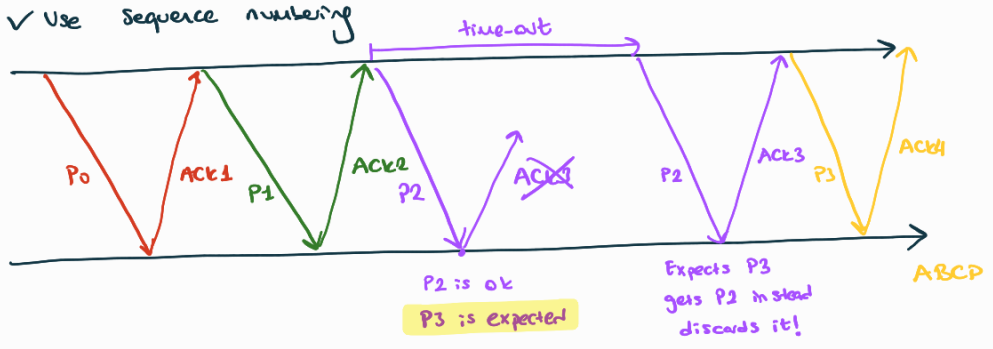
TimeOut:

. Sender keeps timeout for every packet transmitted





✓ Use sequence numbering

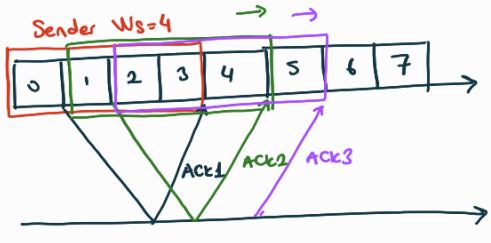


✱ 1 bit sequence number (mod 2) also works

! Stop and Wait is not efficient → instead use pipelining

2) Sliding Window Flow Control

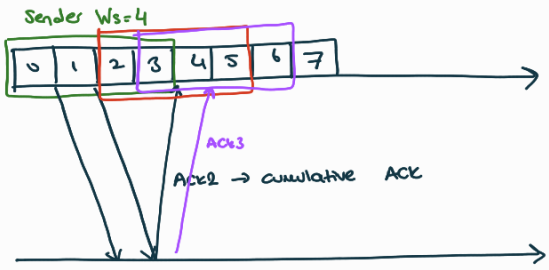
Sender



The sender waits for an ACK before sending another packet

✱ This protocol behaves identically to stop and wait.

Piggybacking Cumulative Acknowledgement



Sequence Number :

Range : $[0 \ldots 2^{k-1}]$

Packets are counted modulo 2^k

How to determine sequence # range?

Receiver

* Progress is independent of the sender window progress (W_R can be $\neq W_S$)

* Any packet which falls outside the receiver window is discarded @ the receiver

What if a data or ACK is lost when using a sliding window protocol ?

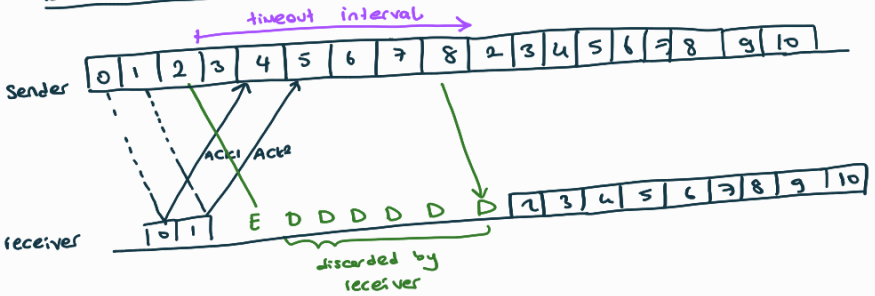
2.1) Sliding Window with Go Back N: ↳ N is the sender window

$W_S > 1$

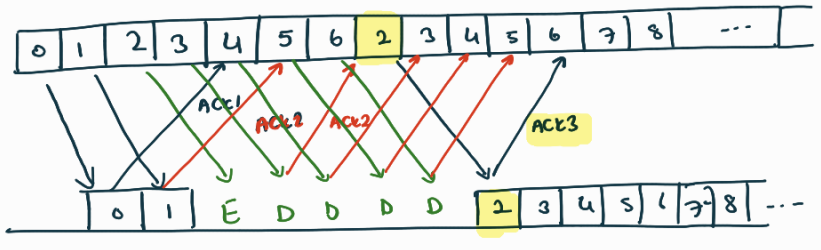
$W_R = 1$

$W_S \leq 2^m - 1$

Retransmission Idea: 1



Retransmission Idea: 2 → duplicate ACK



All frames in the current window are transmitted

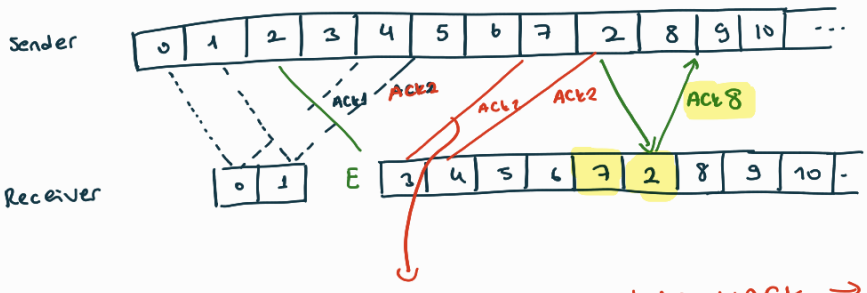
2.2) Sliding Window with Selective Repeat

$W_S > 1$

$W_R > 1$

In general:

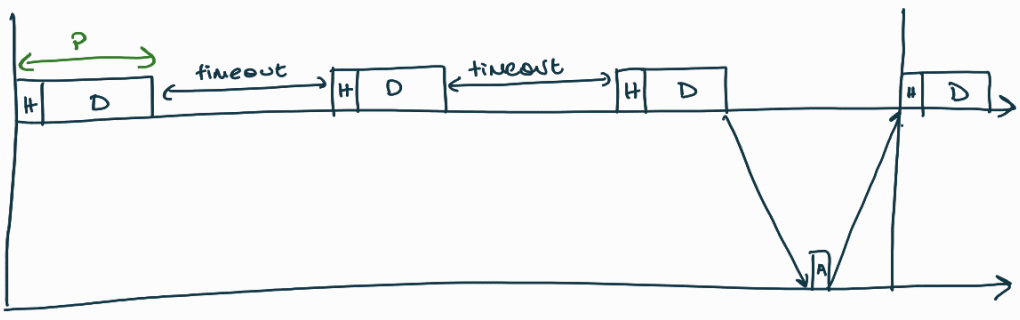
$$W_S = W_R \leq 2^{M-1}$$



Second ACK2 could be NACK → Negative ACK

then it would be renamed as selective repeat with NACK

Performance of Stop-Wait



2 unsuccessful attempts

Last one is successful

$$\eta = \frac{D/b_S}{k \cdot \left(\frac{P}{b_S} + \text{Timeout}\right) + \left(\frac{P}{b_S} + t_{SR} + \frac{A}{b_R} + t_{RS}\right)}$$

k: # of unsuccessful attempts

$k = \frac{L}{1-L}$ → $P_{success}$

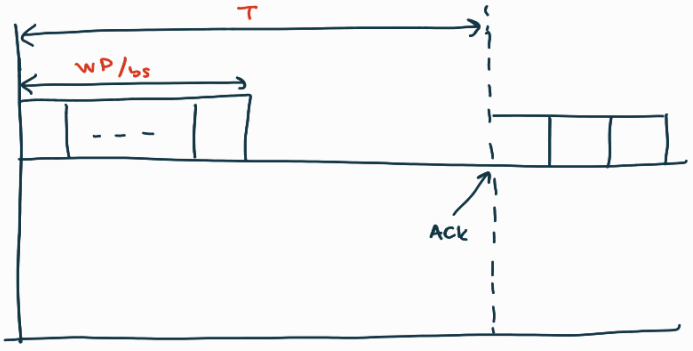
$P_{failure} = 1 - (1-P_1)\cdot(1-P_2)$

P_1: P(loss of data packet)

P_2: P(loss of ACK)

A good timeout = $t_{SR} + \frac{A}{b_R} + t_{RS}$

Performance of Selective Repeat



$$\eta = \left(\frac{D}{D+H}\right)(1-L)\left(\frac{WP/b_S}{T}\right)$$

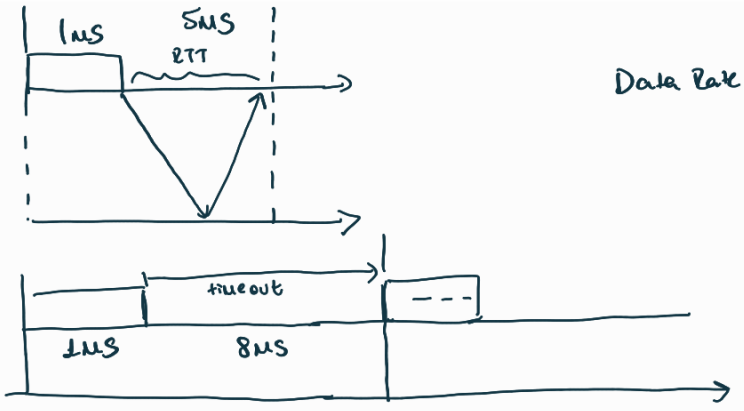
- $\frac{D}{D+H}$: fraction of data
- $(1-L)$: prob of success
- $\frac{WP/b_S}{T}$: frac. of packet transmission in the total time spent

Question 1

$\frac{P}{bs} = \frac{10^3}{10^6 \text{ bits/sec}} = 1 \text{ msec}$ (900 bit data)

A reliable transmission layer has fixed size packets of 1000 bits with 100 bits header. ACK sizes are negligible. The transmission rate is 1Mbps. RTT=5msec. Time out=8msec. The retransmission timer starts when the first bit of the packet is transmitted. The retransmissions are triggered only by time out. The probability of an error in packet transmission or ACK transmission is 0.2. **You can leave your results as fractions with the correct units.**

a) What is the data rate available to the upper layer if stop and wait is used?



$\text{Data Rate} = \frac{\text{Data bits}}{\text{Avg time}} = \frac{900 \text{ bits}}{8 \text{ msec}}$

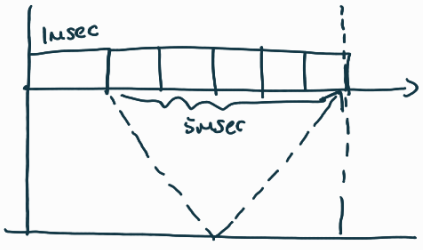
$\text{Expected Retransmission times} = \frac{P_{err}}{1-P_{err}} = \frac{0.2}{0.8} = \frac{1}{4}$

$\text{Total Transmission} = \frac{1}{1-P_{err}} = \frac{1}{0.8} = 1.25$ ($P_{err} \to 0.2$; 0.25 retransmission)

$\text{Avg. time} = 0.25 \times 8 \text{ msec} + 1.6 \text{ msec} = 8 \text{ msec}$

b) Assume that we use sliding window with selective repeat. What should be the sender window size to maximize the data rate available to the upper layer that need the minimum amount of buffer on the sender size? At least how many bits of sequence number is necessary? What is the data rate available in this case?

$W_S \leq 2^m - 1$



$W_S = 6$

$W_S \leq 2^{m-1}$

6

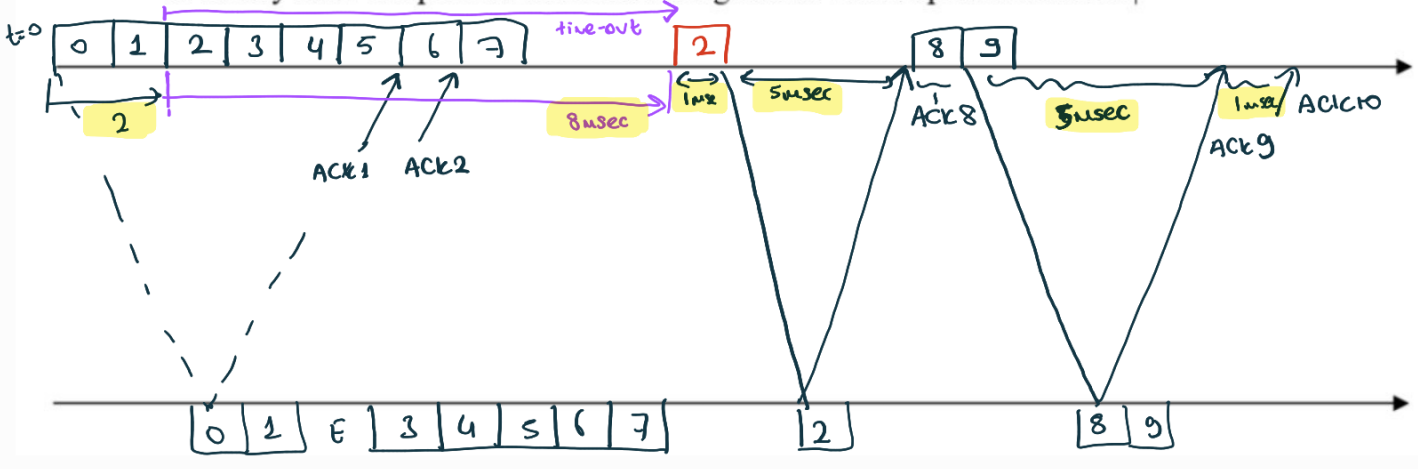
$m - 1 = 3$

$m = 4$

$\text{Data Rate} = 6 \times \frac{900 \text{ bits}}{8 \text{ msec}}$

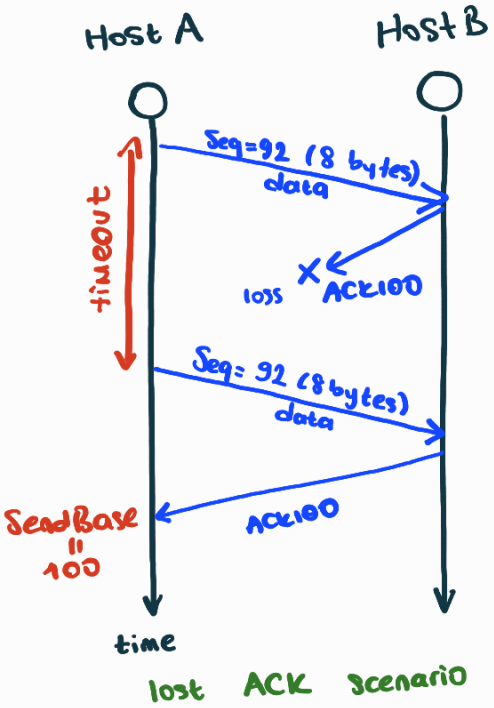
c) Assume that we are using the protocol in part b) with a very large receiver window. The upper layer has 9000 bits to send. The transmission starts at t=0. The third packet (with sequence number 2) is in error. What is the required time to complete this transmission? What is the available data rate for the upper layer for this transmission? Complete the timing diagram below to clearly show the packets and acknowledgements with sequence numbers.

10 packets

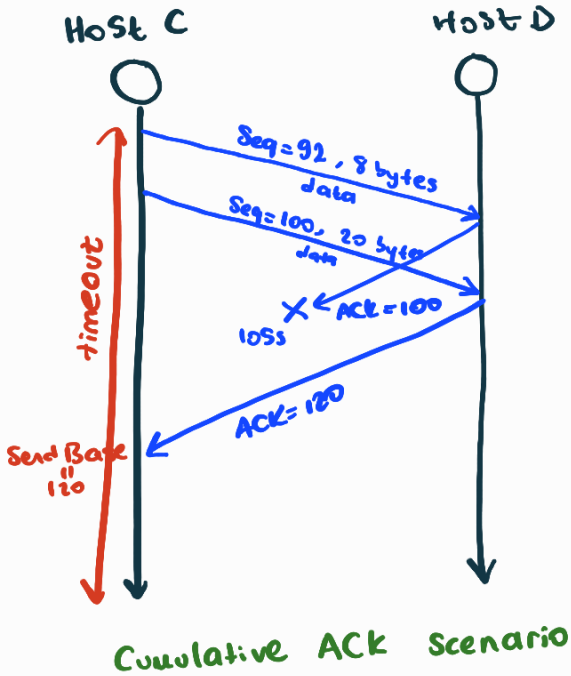


$\text{Data Rate} = \frac{9000 \text{ bits}}{2+8+6+6+1} = \frac{9000 \text{ bits}}{23 \text{ msec}}$

TCP: Retransmission Scenarios



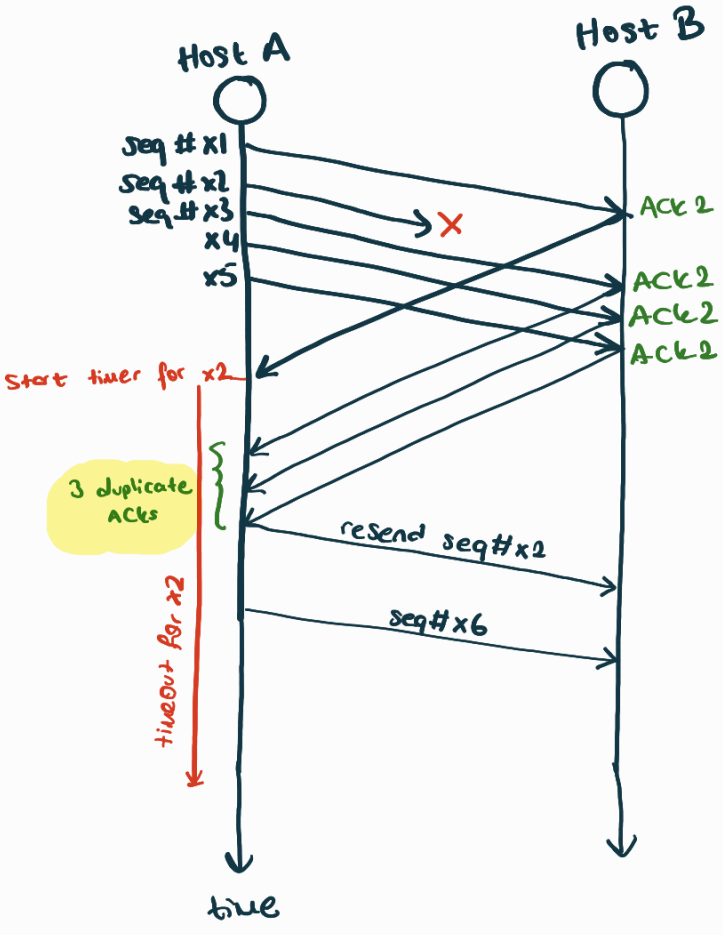
lost ACK scenario



Cumulative ACK Scenario

Fast Retransmit

- time-out period often relatively long.
- detect lost segments via duplicate ACKs



TCP ACK Generation

Case 1:

Event @ Receiver	TCP Receiver Action
Arrival of in-order segment with expected seq #. All data up to expected seq # already ACKed.	Delayed ACK. Wait up to 500 ms for next segment. If no next segment, send ACK

Ex Initially: Receiver window = [100-199]

t = 0 ms: Receive segment [100-109]

No new segment

✓Send ACK 110 @ t = 150 ms (delayed ACK)

Case 2:

Expected seq #. One other segment has ACK pending.	Immediately send single cumulative ACK

t=0 ms: Receive segment [100-109]

t=20 ms: Receive segment [110-119]

Send immediate cumulative ACK.

Case 3

Arrival of out-of-order segment higher-than-expect seq#	Immediately send ACK, indicated seq. # of next expected byte

t=0 ms: Receive segment [100-109]

t=120 ms: No new segment. send ACK 110 (delayed ACK)

t=140 ms: Receive segment [120-129]

Send immediate duplicate ACK 110 (Case 3)

t=500 ms: Receive segment [110-115]

Send immediate ACK 116 (Case 4)

↙

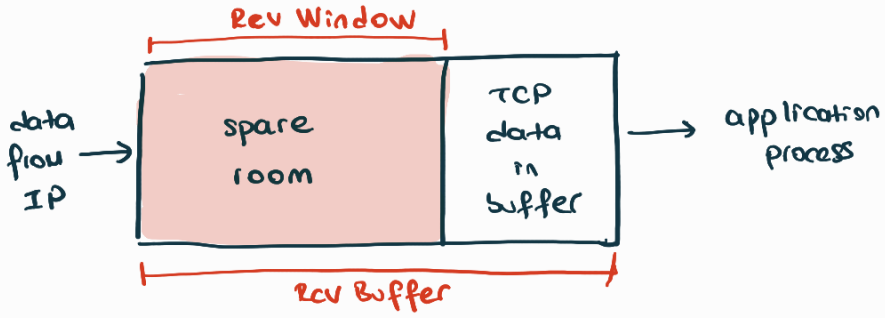
When gap is detected

TCP Flow and Congestion Control

- Low Capacity Receiver → rwnd
- Too much network load, high queuing delays, packet drops → cwnd

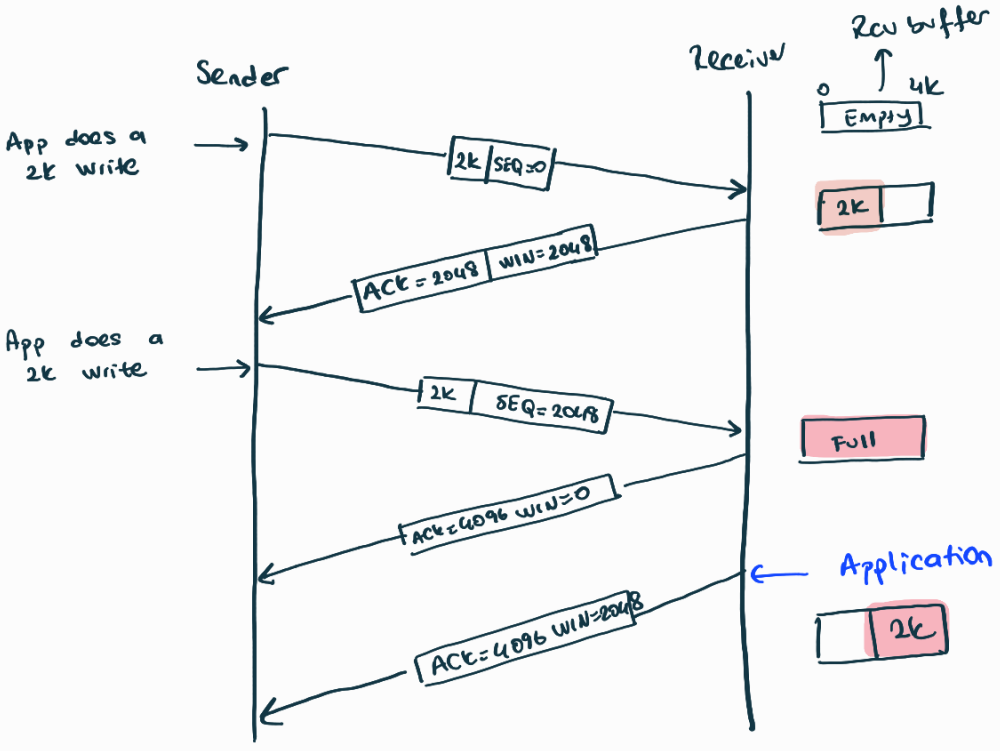
Sender window = min(cwnd, rwnd)

(Sender) Rate = $\frac{W \text{ (byte)}}{RTT \text{ (secs)}}$



RcvWindow = rcvwnd = RcvBuffer − [LastByteRcvd − LastByteRead]

(LastByteRcvd − LastByteRead: TCP data in buffer)



Seq #s:

byte stream number of first byte in segment's data.

ACKs:

seq # of next byte expected from other side (cumulative ACK)

TCP Congestion Control

Goal: TCP sender should transmit as fast as possible, but without congesting network

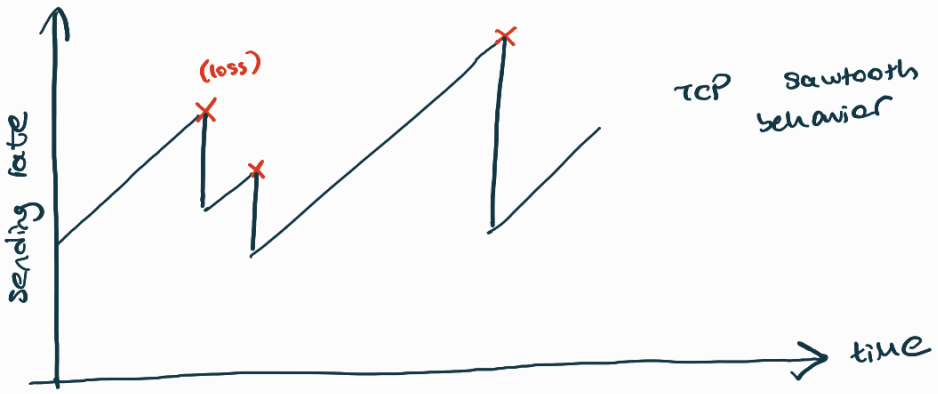
Last Byte Sent - Last Byte Acked ≤ min (cwnd, rwnd)

End-to-end congestion control:

→ ACK: segment received, network not congested, so increase sending rate

→ Loss Event: timeout or 3 duplicate ACKs.

probing for bandwidth



Mechanisms

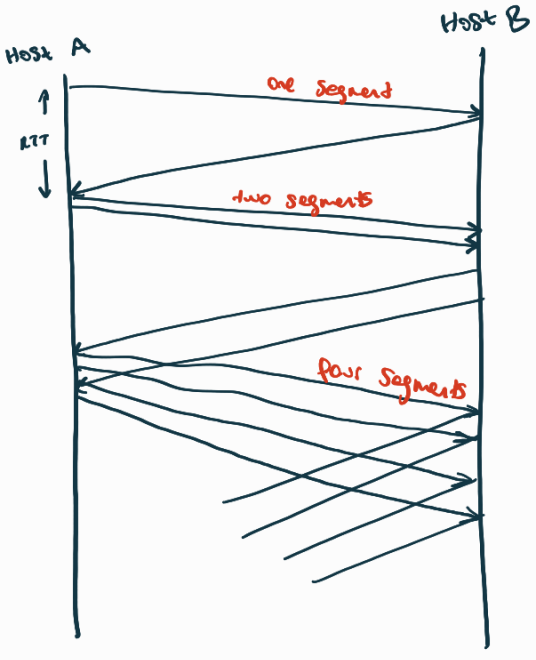
→ Initial Rate Increase

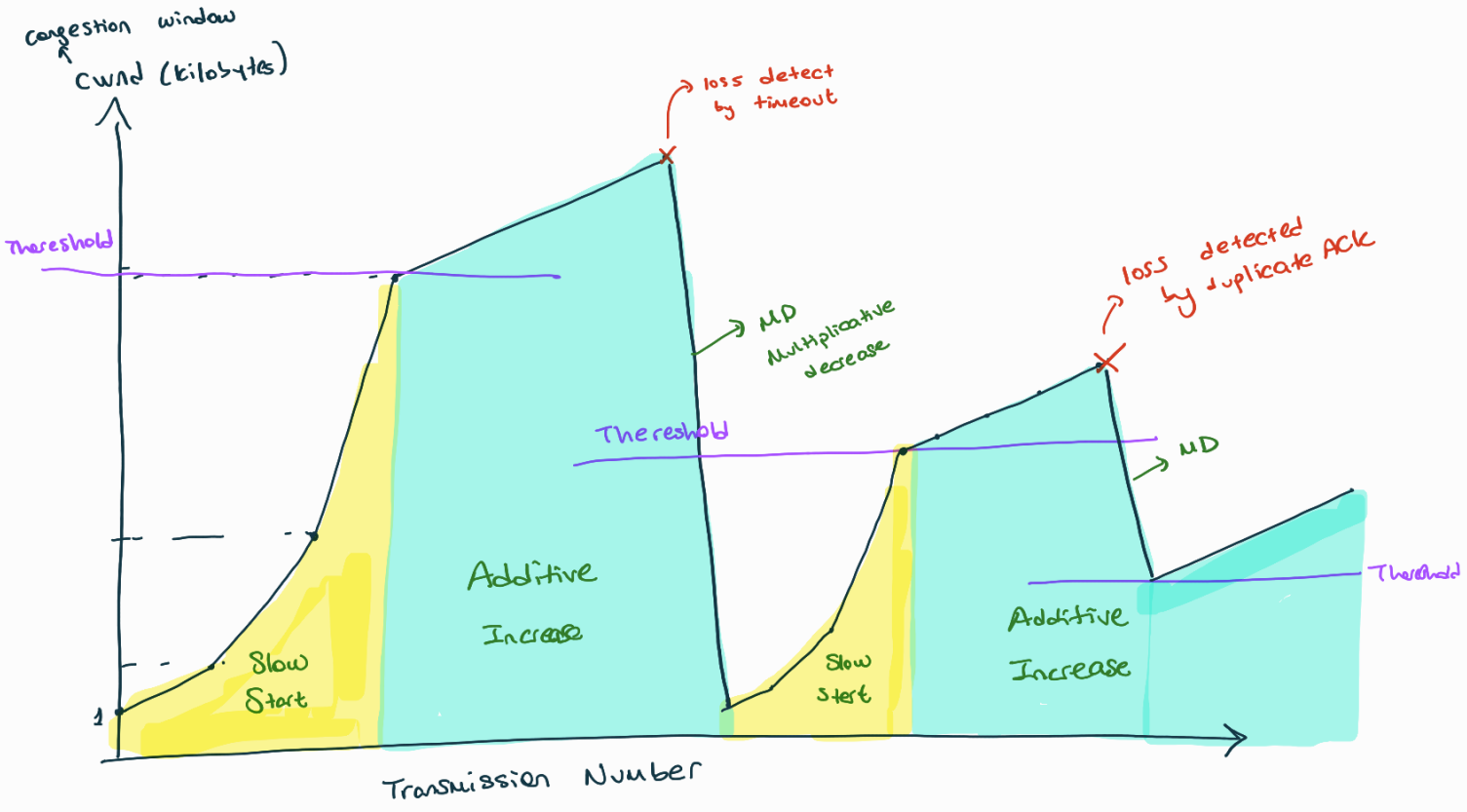
- Slow Start (exponential increase until threshold)
- Congestion Avoidance (linear increase after threshold)

window size to switch from exp. increase to linear increase

→ Fluctuating around the available rate

- Additive Increase Multiple Decrease (AIMD)





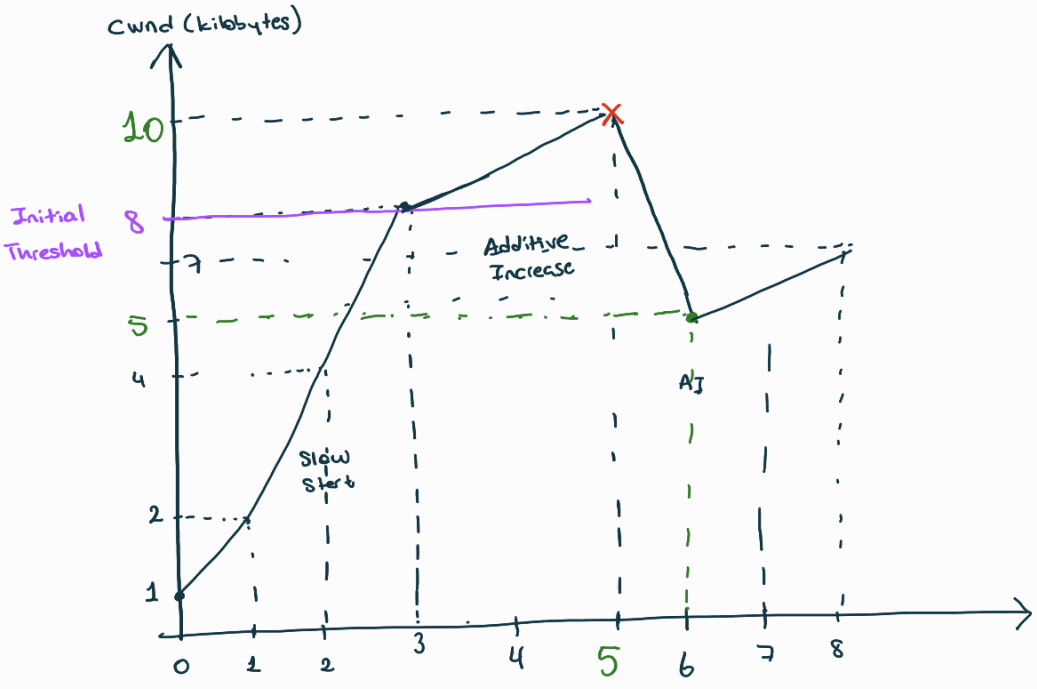
Questions

Q1)

Draw the congestion window vs RTT graph for a TCP connection (TCP RENO) starting from RTT=0 until RTT= 8 (including RTT=8) according to the following information:

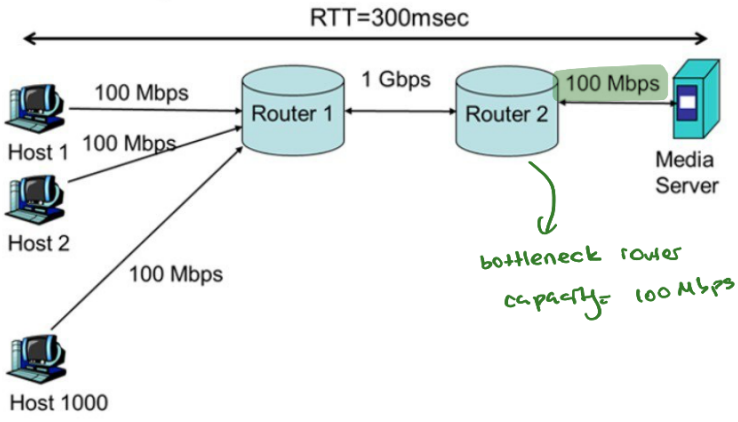
- Initial Threshold=8
- A Triple-ACK occurs at RTT=6. This affects the size of the window sent at RTT=6.

Indicate the window size for RTT=6 and change of threshold. Complete the graph until RTT=8 (including RTT=8).



Question 2

Consider the configuration below.



The media server is providing content to 1000 Hosts. The link bitrates are indicated in the figure. The media files are 1 Mbits long. The hosts continuously download these media files using persistent HTTP. There is no base HTML file.

a) What is the average data rate achieved by each Host at the application layer? Which TCP property you use for this computation?
b) Assuming that the slow start finishes quickly at which window size in bits does the media server get a triple acknowledgement from any given host?
c) What is the average total time for a client to download one media file?

a) Average Rate = $\frac{100 \text{ Mbps}}{1000}$ = 0.1 Mbps per host → avg. throughput

TCP Fairness

b) $\left(\frac{3}{4}\right)\frac{W}{RTT} = 0.1 \text{ Mbps}$

$\frac{3}{4}\frac{W}{300 \text{ msec}} = 0.1 \times 10^6 \text{ bit/sec}$

$W = \frac{4 \times 300 \times 10^{-3} \text{ sec} \times 0.1 \times 10^6}{3}$

$W = 40 \text{ Kbits}$

c) Persistent HTTP

↓

RTT + N * (RTT + transmit time)

↑ 1 object

$300 \text{ msec} + 300 \text{ msec} + \frac{1 \text{ Mbit}}{0.1 \text{ Mbit/sec}} = 10.6 \text{ sec}$ → avg. total time

10 sec

Question 3

Assume that a TCP receiver delays ACKs by a fixed delay of 100msec when it can. Right before t=0 TCP receiver sent all ACKs for the previously segments. The receiver window is [100-199].

At t=0 receiver gets segment with sequence number=100, size=20 Bytes.

At t=90 msec receiver gets segment with sequence number=120, size= 20 Bytes.

At t=120 msec receiver gets segment with sequence number=160, size=20 Bytes.

At t=130 msec receiver get segment with sequence number=140, size =20 Bytes.

Time ACK sent	ACK Sequence Number	Reason
t=90	120+20=140	New Segment
t=120	160	Gap
t=130	180	Gap Fill