2025 Computer Networks
ITM 21102052 Jeong-yun LEE

Chapter 3 Problems

Problem 27. Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

a. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?

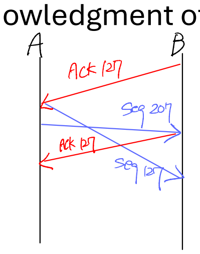
Seq #: 127 + 80 = 207
src port #: 302
dst port #: 80

b. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?

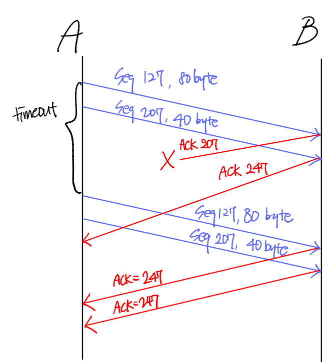
ACK #: 207
Src port #: 80
dst port #: 302

c. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

ACK #: 127



d. Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.



2025 Computer Networks
ITM 21102052 Jeong-yun LEE

Problem 28. Host A and B are directly connected with a 100 Mbps link. There is one TCP connection between the two hosts, and Host A is sending to Host B an enormous file over this connection. Host A can send its application data into its TCP socket at a rate as high as 120 Mbps but Host B can read out of its TCP receive buffer at a maximum rate of 50 Mbps. Describe the effect of TCP flow control.

Host B will advertise its free buffer space by "rwnd".
So, host A will slow down its transfer speed to prevent overflow of host B's buffer depending on rwnd value B sent.

Problem 32. Consider the TCP procedure for estimating RTT. Suppose that $\alpha = 0.1$. Let *SampleRTT1* be the most recent sample RTT, let *SampleRTT2* be the next most recent sample RTT, and so on.

a. For a given TCP connection, suppose four acknowledgments have been returned with corresponding sample RTTs: *SampleRTT4, SampleRTT3*, *SampleRTT2*, and *SampleRTT1*. Express *EstimatedRTT* in terms of the four sample RTTs.

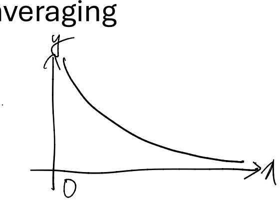
$$EstimatedRTT_1 = SampleRTT_1$$
$$EstimatedRTT_2 = (1-0.1)EstimatedRTT_1 + (0.1) \times SampleRTT_2 = (1-0.1)SampleRTT_1 + (0.1)SampleRTT_2$$
$$EstimatedRTT_3 = (1-0.1)EstimatedRTT_2 + (0.1)SampleRTT_3 = (1-0.1)\{(1-0.1)SampleRTT_1 + (0.1)SampleRTT_2\} + (0.1)SampleRTT_3$$
$$EstimatedRTT_4 = (1-0.1)EstimatedRTT_3 + (0.1)SampleRTT_4 = (1-0.1)\{(1-0.1)^2 SampleRTT_1 + (1-0.1)(0.1)SampleRTT_2 + (0.1)SampleRTT_3\} + (0.1)SampleRTT_4$$

b. Generalize your formula for *n* sample RTTs.

$$EstimatedRTT_n = \sum_{k=1}^{n} \alpha(1-\alpha)^{n-k} \cdot SampleRTT_k \quad , \quad \alpha = 0.1$$

c. For the formula in part (b) let *n* approach infinity. Comment on why this averaging procedure is called an exponential moving average.

$$\lim_{n\to\infty} \sum_{k=1}^{n} \alpha(1-\alpha)^{n-k} SampleRTT_k \Rightarrow$$ old values' effect of entire result is exponential decay. but recent values' effect of the result is weighted more.



2025 Computer Networks
ITM 21102052 Jeong-yun LEE

Problem 34. What is the relationship between the variable *SendBase* in **Section 3.5.4** and the variable *LastByteRcvd* in **Section 3.5.5** ?

SendBase $-1 \leq$ LastByteRcvd

SendBase of sender is the oldest seq # not ACKed

LastByteRcvd of receiver is the recent byte received.

In normal TCP communication, Sendbase -1 is last seq # of last byte Acked.
It cannot bigger than LastByteRcvd, because sender cannot wait for ACK already ACKed.

Problem 35. What is the relationship between the variable *LastByteRcvd* in **Section 3.5.5** and the variable y in Section **3.5.4**?

$y \leq$ LastByteRcvd $+ 1$

y is ACK # from receiver.

So, reciever can make ACK # until LastByteRcvd.

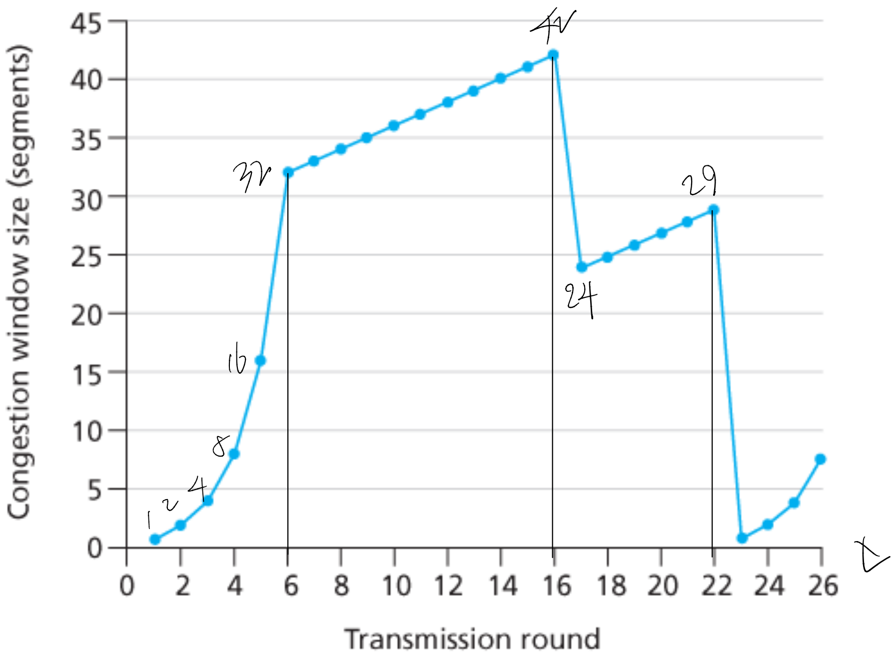
Thus y cannot be bigger than LastByteRcvd $+ 1$

Problem 36. In **Section 3.5.4,** we saw that TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first duplicate ACK for a segment is received?

I think the reason why fast retransmit is not executed before 3 duplicate ACKs is ~~to~~ wait for the missing ACKs to arrive.

If retransmission is performed for all duplicate ACKs, performance overhead may become excessive.

Problem 40. Consider **Figure 3.58.** Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.



a. Identify the intervals of time when TCP slow start is operating.

t = 1 ~ 6 , 23 ~ 26

congestion window size is increased exponentially.

b. Identify the intervals of time when TCP congestion avoidance is operating.

t = 6 ~ 16 , 17 ~ 22

cwnd increases linearly

c. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

by triple duplicate ACKs.

If loss detected by timeout, then cwnd would be set to 1.

d. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

by timeout

because cwnd is dropped to 1.

2025 Computer Networks
ITM 21102052 Jeong-yun LEE

e. What is the initial value of *ssthresh* at the first transmission round?

32

After 32 segment, the cwnd increases linearly.

f. What is the value of *ssthresh* at the 18th transmission round?

21

At round 16, cwnd is 42. So, 42/2 = 21 is set to ssthresh due to triple duplicate ACKs

g. What is the value of *ssthresh* at the 24th transmission round?

14.

At round 22, cwnd = 29.
after decreasing of cwnd, ssthresh is set to $\lfloor 29/2 \rfloor = 14$

h. During what transmission round is the 70th segment sent?

1st = 1 round
2nd–3rd = 2 round
4th–7th = 3 round
8–15th = 4 round
16–31 = 5 round
32–63 = 6 round
64–127 = 7 round

↳ 70th segment is sent at round 6.

i. Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of *ssthresh*?

ssthresh = 8/2 = 4
cwnd = 4+3 = 7
26 round cwnd = 8.

j. Suppose TCP Tahoe is used (instead of TCP Reno) and assume that triple duplicate ACKs are received at the 16th round. What are the *ssthresh* and the congestion window size at the 19th round?

ssthresh = 42/2 = 21
cwnd = 4 (17th = 1 18th = 2 19th = 4)

k. Again, suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

17th = 1
18th = 2
19th = 4
20th = 8
21th = 16
22th = 21

Σ = 52 packets.

8 15
37