1.

P26. Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 536 bytes.

a. What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.

b. For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

a)

The size of TCP sequence number field = 4 bytes

=4*8 bits

=32 bits.

So, the sequence numbers of bits are 2^{32}

The maximum file size sent from Host A to Host B representable by 2^{32}

$= 2^2 \times 2^{30}$ bytes

$= 2^2$ Gbytes

$= 4$ Gbytes

b)

Maximum segment size (MSS) = 536 bytes

Segments data = $2^{32}/536$

= 8012999

Total header fields = 66 bytes.

Total number of bytes through the 155Mbps link = 8012999×66 bytes

= 528857934 bytes

Transmitted data = $(2^{32} + 528857934)$

$= 4.824 \times 10^9$ bytes

$$\text{Transmit time} = \frac{4.824 \times 10^9 \times 8\text{bits}}{155 \times 10^6\ \text{bps}} \approx \boxed{249}\ \text{seconds}$$

2.

P27. 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?

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?

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

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.

Given data:

- Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126.
- The first and second segments contain 80 and 40 bytes of data, respectively.
- The first segment of sequence number is 127.
- The source port number is 302.
- The destination port number is 80

a)

Sequence number = first segment of sequence number+ destination port number

=127+80

=207

So, sequence number=207

Source port number = 302

Destination port number= 80

b)

Acknowledgement number= 207

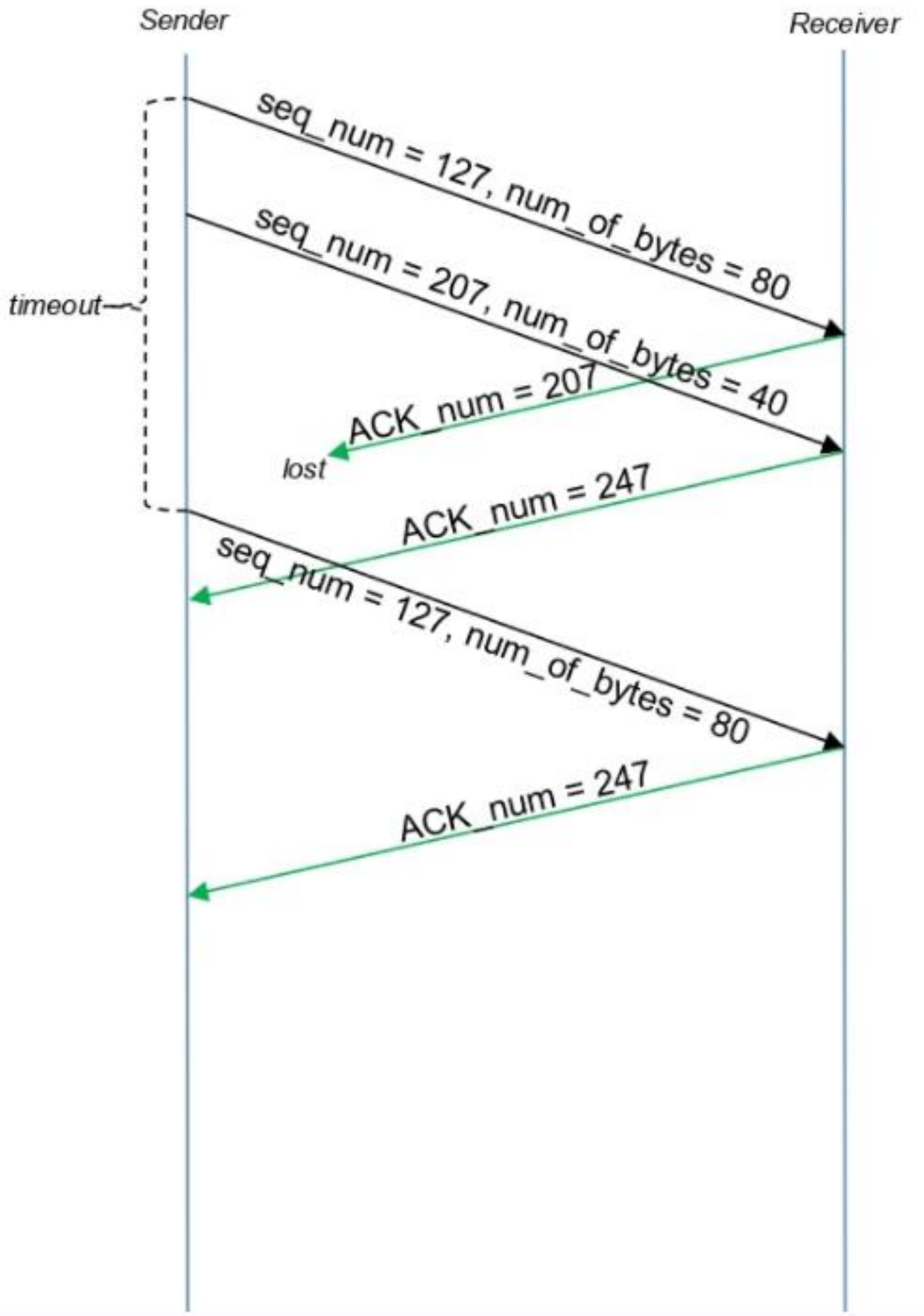
Source port number = 80

Destination port number= 302

c)

Acknowledgement number=127

d) Timeline diagram



3.

P37. Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host A sends 5 data segments to Host B, and the 2nd segment (sent from A) is lost. In the end, all 5 data segments have been correctly received by Host B.

a. How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.

b. If the timeout values for all three protocol are much longer than 5 RTT, then which protocol successfully delivers all five data segments in shortest time interval?

a)

Consider given data:

For *GBN*:

- They are primarily A sent 5 segments: 1, 2, 3, 4, 5 and then later re sent 4 segments: 2, 3, 4 and 5 later.
- So, total A sends (5+4)= 9 segements.
- **A sends 9 segments: 1 2 3 4 5 2 3 4 5**
- B sent **4** ACKs with sequence number of 1s(1 1 1 1) and **4** ACKs with sequence numbers 2, 3, 4 and 5. So total B sends (4+4)=**8** ACKs.
- **B sends 8 ACKs: 1 1 1 1 2 3 4 5**

For *SR*:

- A sent 5 segments: 1, 2, 3, 4, 5 and then later re sent only one segment: 2.
- So, total A sends (5+1)= 6 segements.
- **A sends 6 segments: 1 2 3 4 5 2**
- B sent 4 ACKs with sequence number 1, 3, 4, 5 and there is only one ACK with sequence number 2.
- So, total B sends (4+1)= 5 ACKs.
- **B sends 5 ACKs: 1 3 4 5 2**

For *TCP*:

- A sent 5 segments: 1, 2, 3, 4, 5 and then later re sent only one segment: 2.
- So, total A sends (5+1)= 6 segements.
- **A sends 6 segments: 1 2 3 4 5 2**
- B sent 4 ACKs with sequence number 2(2 2 2 2) and there is only one ACK with sequence numbers 6.
- So, total B sends (4+1)= 5 segements.
- **B sends 5 ACKs: 2 2 2 2 6**

b)

If the timeout values for all three protocol are much longer than 5 RTT, then ***TCP*** protocol successfully delivers all five data segments in shortest time interval. The reason is that TCP uses fast retransmit without waiting time.

4.

P40. 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.

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

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

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

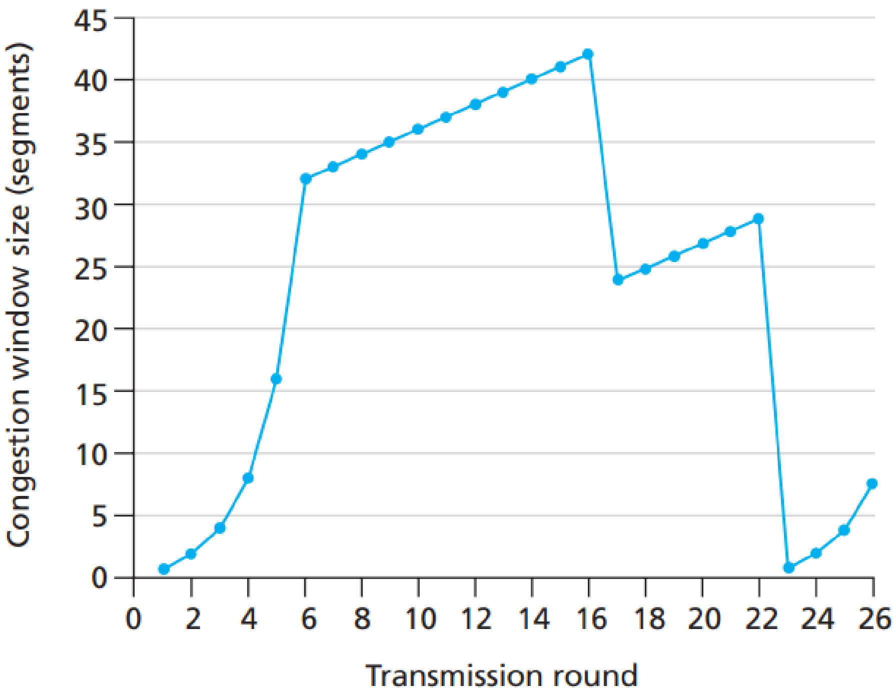


Figure 3.58 ♦ TCP window size as a function of time

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

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

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

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

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`?

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?

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?

a) If TCP slow start is operating, then the intervals of time **1 to 6 and 23 to 26**.
b) If TCP congestion avoidance is operating, then the intervals of time **6 to 23**.
c) After the 16th transmission round, then the segment loss detected by **a triple duplicate ACK**.
d) After the 22nd transmission round, then the segment loss detected by **timeout**.
e) The initial value of ssthresh at the first transmission round **32**.
f) The value of ssthresh at the 18th transmission round **21**.
g) The value of ssthresh at the 24th transmission round **13**.
h) The transmission round is the 70th segment sent is **7**.
i) If a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, then the value is **4**.
j) Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. Then the ssthresh and the congestion window size at the 19th round is 1 and transmission round is 21.
k) Again suppose TCP Tahoe is used, and there is a timeout event at 22^{nd} round, then the packets have been sent out from 17th round till 22^{nd} round(inclusive) is **52**.

5.

P41. Refer to Figure 3.55, which illustrates the convergence of TCP's AIMD algorithm. Suppose that instead of a multiplicative decrease, TCP decreased the window size by a constant amount. Would the resulting AIAD algorithm converge to an equal share algorithm? Justify your answer using a diagram similar to Figure 3.55.

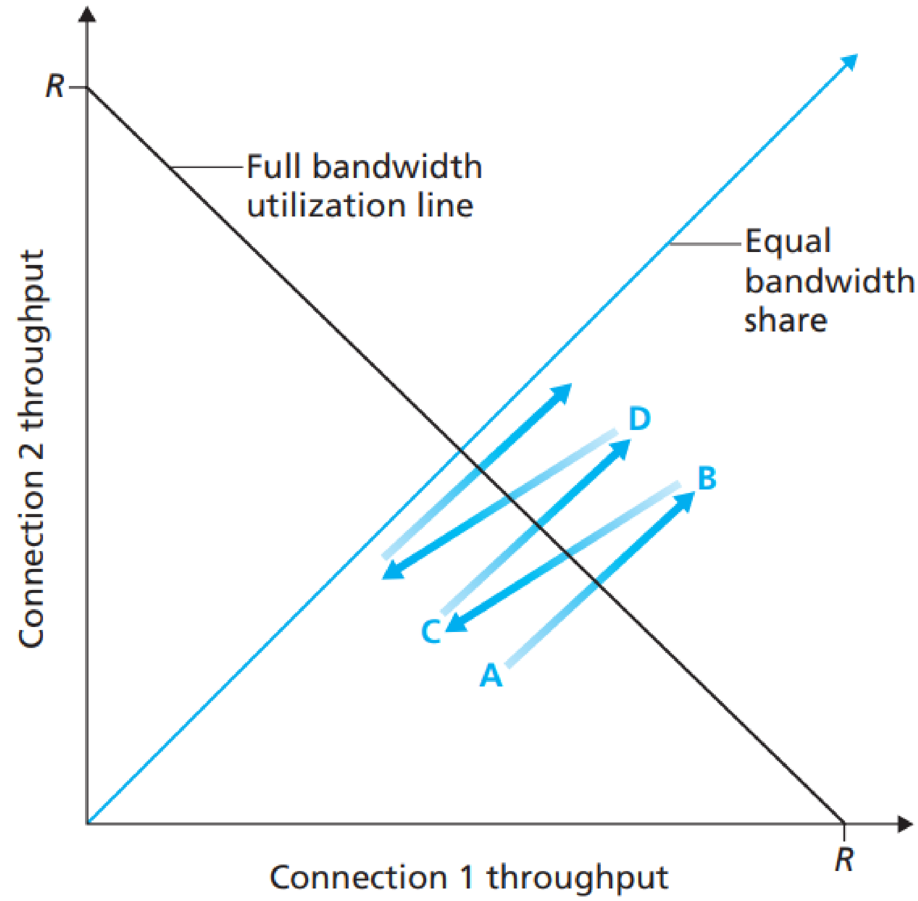
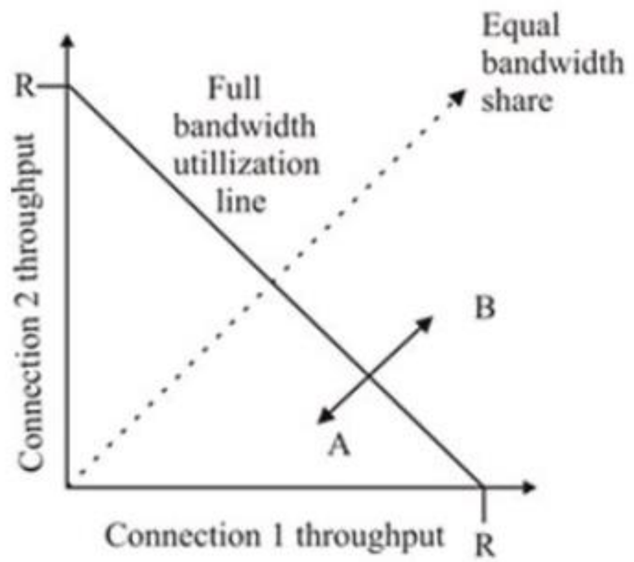


Figure 3.55 ♦ Throughput realized by TCP connections 1 and 2

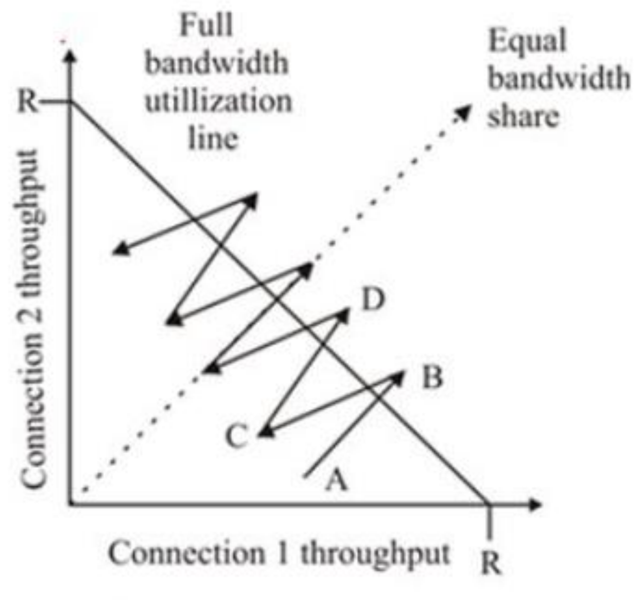
Refer to Figure 3.56 from the text book:

Consider given data:
Suppose that instead of a multiplicative decrease, TCP decreased the window size by a constant amount.

The result diagram is as follows:



The loss ratio of the linear *decrease* is 1. Similarly, the loss ratio of the linear *increase* is 1. Both the ratios are same. So, from the line segment $\overline{AB}$, the throughput never moves off.



(b) linear increase, connection 1 decrease is twice that of connection 2

The loss ratio of the linear decrease on loss between the connections is 2:1.
So, the full link bandwidth is 2.