HW2 Review

P4. Calculate the Internet checksum.

a. Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1's complement of the sum of these 2 bytes?

```
        01011100
       +01100101
       =11000001
1’s c:00111110
```

b. Suppose you have the following 2 bytes: 11011010 and 01100101. What is the 1's complement of the sum of these 2 bytes?

```
        11011010
      + 01100101
      =100111111
  wrap:01000000
1’s c:10111111
```

c. For the bytes in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1's complement doesn't change.

```
       01011100
      +01100101
      =11000001
1’s c:00111110
```

P14. Consider a reliable data transfer protocol that uses only negative acknowledgments. Suppose the sender sends data only infrequently. Would a NAK-only protocol be preferable to a protocol that uses ACKs? Why? Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

In a NAK only protocol, the loss of packet x is only detected by the receiver when packet x+1 is received. That is, the receivers receives x-1 and then x+1, only when x+1 is received does the receiver realize that x was missed. If there is a long delay between the transmission of x and the transmission of x+1, then it will be a long time until x can be recovered, under a NAK only protocol.

On the other hand, if data is being sent often, then recovery under a NAK-only scheme could happen quickly. Moreover, if errors are infrequent, then NAKs are only occasionally sent (when needed), and ACK are never sent – a significant reduction in feedback in the NAK-only case over the ACK-only case.

P15. Consider the cross-country example shown in Figure 3.17 . How big would the window size have to be for the channel utilization to be greater than 98 percent? Suppose that the size of a packet is 1,500 bytes, including both header fields and data.

Utilization = (nL/R)/(L/R + RTT)

L/R = 1500 bytes / 1 Gbps = $1500*8 / 10^9 = 1.2*10^{-6}$ s = 0.012 ms

Utilization = n 0.012 / (0.012 + 30) >= 98%, so n >= 2451.

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?

In the second segment from Host A to B, the sequence number is 207, source port number is 302 and destination port number is 80.

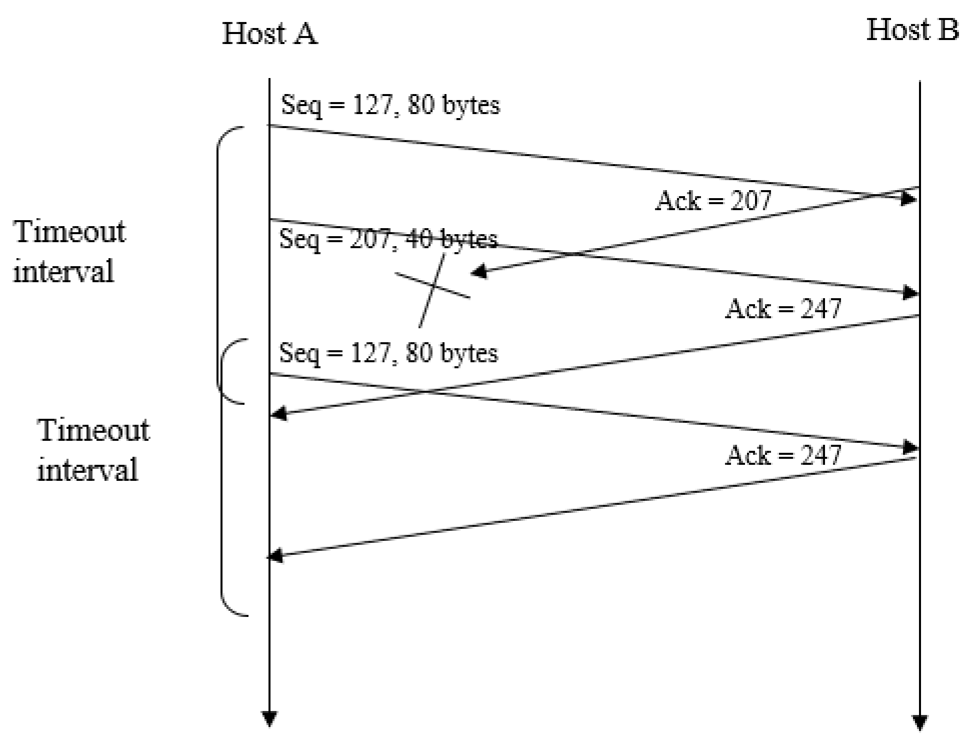
b. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?

If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 207, the source port number is 80 and the destination port number is 302.

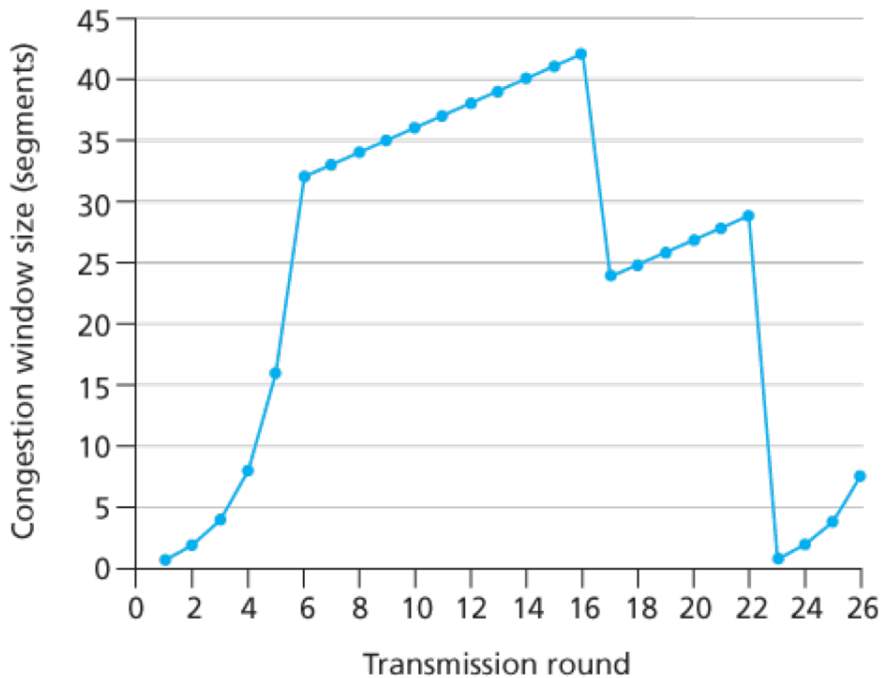
c. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, the acknowledgement number is 127, indicating that it is still waiting for bytes 127 and onwards.

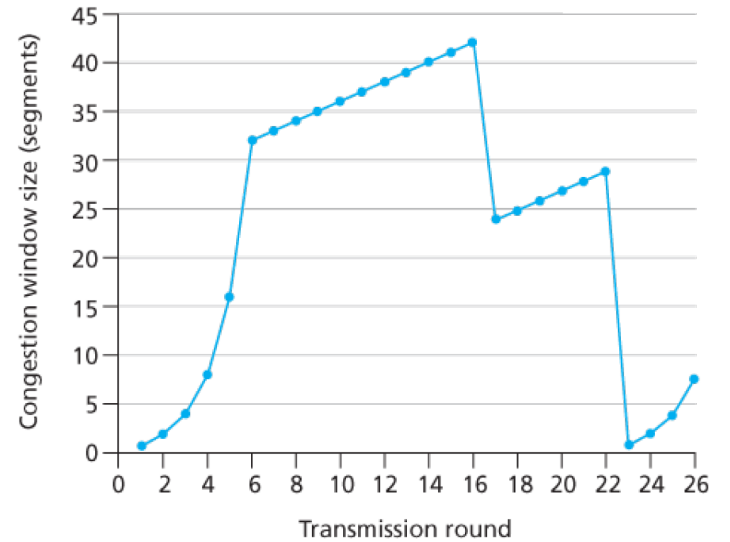
d. Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) 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.



P40. Consider the following figure. Assuming TCP Reno is the protocol experiencing the behavior shown, 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.



TCP slowstart is operating in the intervals [1,6] and [23,26].

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

TCP congestion avoidance is operating in the intervals [6,16] and [17,22]

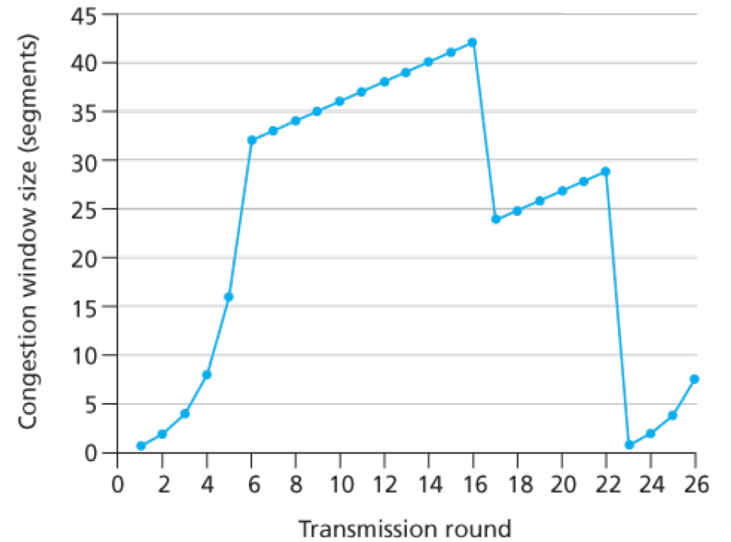
c. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

After the 16th transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.

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

After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.

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



The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.

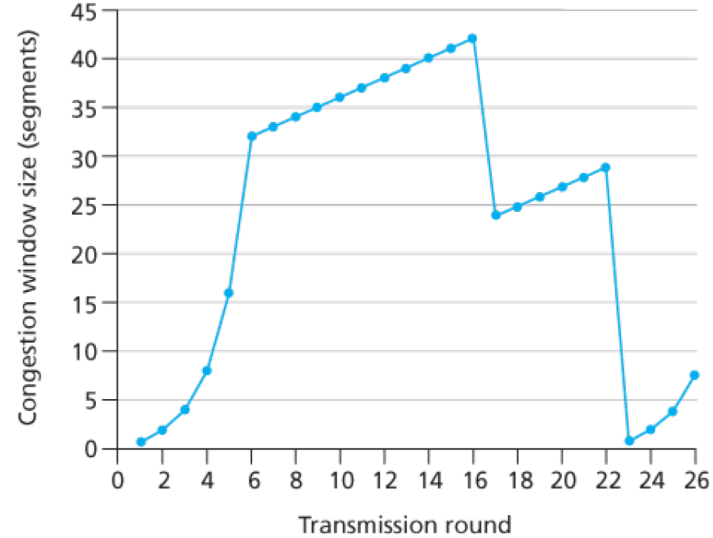
f. What is the value of ssthresh at the 18th transmission round?

The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.

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

The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion windows size is 29. Hence the threshold is 14.5 during the 24th transmission round.

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

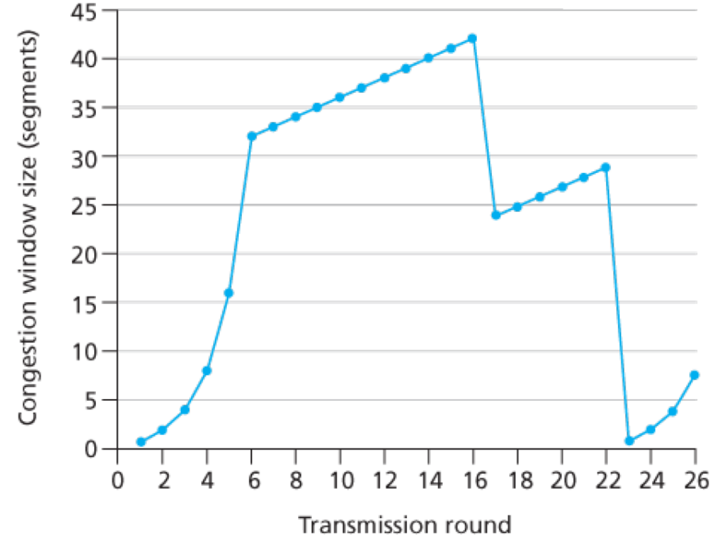


During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets 16-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets 64 – 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.

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?

The threshold will be set to half the current value of the congestion window (8) when the loss occurred and congestion window will be set to the new threshold value + 3 MSS . Thus the new values of the threshold and window will be 4 and 7 respectively.

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?



Round 17: ssthresh = 42/2 =21 cwnd = 1
Round 18: ssthresh = 21, cwnd = 2
Round 19: ssthresh = 21, cwnd = 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?

Round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.

P46. Consider that only a single TCP (Reno) connection uses one 10Mbps link which does not buffer any data. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver's receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two-way propagation delay of this connection is 150 msec; and this TCP connection is always in congestion avoidance phase, that is, ignore slow start.

a. What is the maximum window size (in segments) that this TCP connection can achieve?

Let W denote the max window size measured in segments. Then, W*MSS/RTT = 10Mbps, as packets will be dropped if the maximum sending rate exceeds link capacity. Thus, we have W*1500*8/(0.15+1500*8/10^7)=10*10^6, then W is about 126 segments.

b. What is the average window size (in segments) and average throughput (in bps) of this TCP connection?

As congestion window size varies from W/2=63 to W=126, then the average window size is 0.75W=94.5 segments. Average throughput is 94*1500*8/0.15 =7.5Mbps.

c. How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?

126/2 *0.15 = 9.45 seconds, as the number of RTTs (that this TCP connections needs in order to increase its window size from W/2 to W) is given by W/2. Recall the window size increases by one in each RTT.