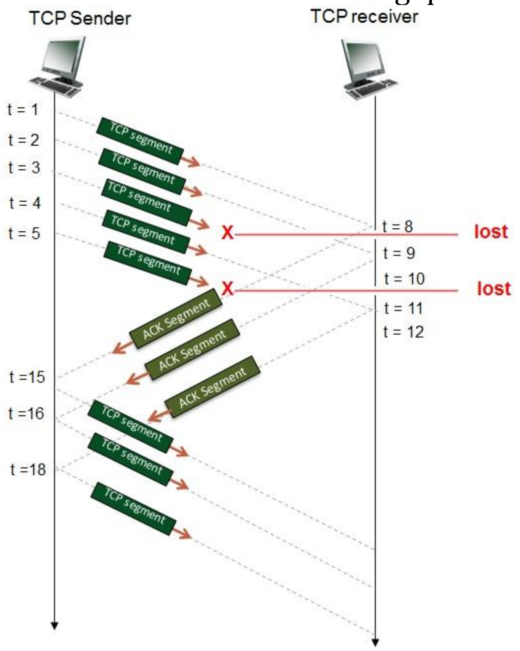
Homework (5 points)

1. Computing an Internet checksum (1 point)

Consider the two 16-bit words (shown in binary) below. Recall that to compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum [1] of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result. Compute the Internet checksum value for these two 16-bit words:

01100110 10011111 this binary number is 26271 decimal (base 10)
10101010 11010001 this binary number is 43729 decimal (base 10)

2. Consider the figure below in which a TCP sender and receiver communicate over a connection in which the sender-to-receiver segments may be lost. The TCP sender sends initial window of five segments at t=1,2,3,4,5, respectively. Suppose the initial value of the sender-to-receiver sequence number is 110 and all sender-to-receiver segments each contain 594 bytes. The delay between the sender and the receiver is 7 time units, and so the first segment arrives at the receiver at t=8, and a receiver-to-sender ACK for this segment arrives at the TCP sender at t=15. As shown in the figure, 2 of the five segments are lost between the sender and the receiver, but none of the receiver-to-sender ACKs are lost. Answer the following questions: (1 point)

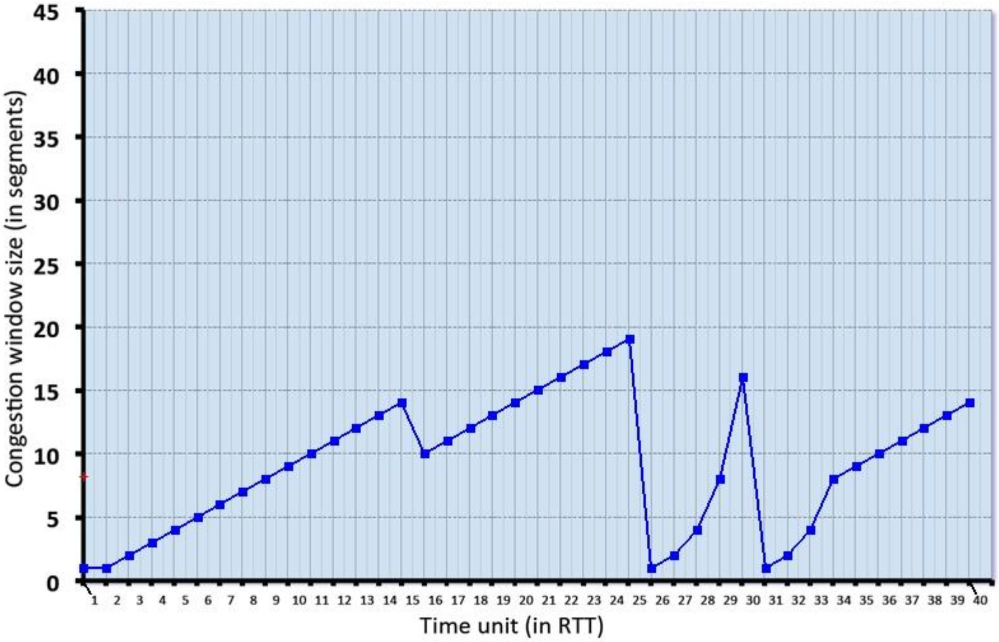


Give the sequence number field value in the sender-to-receiver segments in the first set of segments that are sent beginning at t=1, and the acknowledgement number field value in the receiver-to-sender ACK segments that are sent from the receiver back to the sender beginning at t=8.

Next consider the sender-to-receiver segments generated by the sender in response to the ACKS that are received beginning at t=15. Give the sequence number of the segments transmitted and a brief explanation of why a given segment is transmitted, or not transmitted, on receipt of an ACK.

3. **TCP in action: slow start, congestion avoidance, and fast retransmit.** (2 points)

Consider the figure below, which plots the evolution of TCP's congestion window at the beginning of each time unit (where the unit of time is equal to the RTT); see Figure 3.53 in the text. In the abstract model for this problem, TCP sends a "flight" of packets of size cwnd at the beginning of each time unit. The result of sending that flight of packets is that either (i) all packets are ACKed at the end of the time unit, (ii) there is a timeout for the first packet, or (iii) there is a triple duplicate ACK for the first packet. In this problem, you are asked to reconstruct the sequence of events (ACKs, losses) that resulted in the evolution of TCP's cwnd shown below.

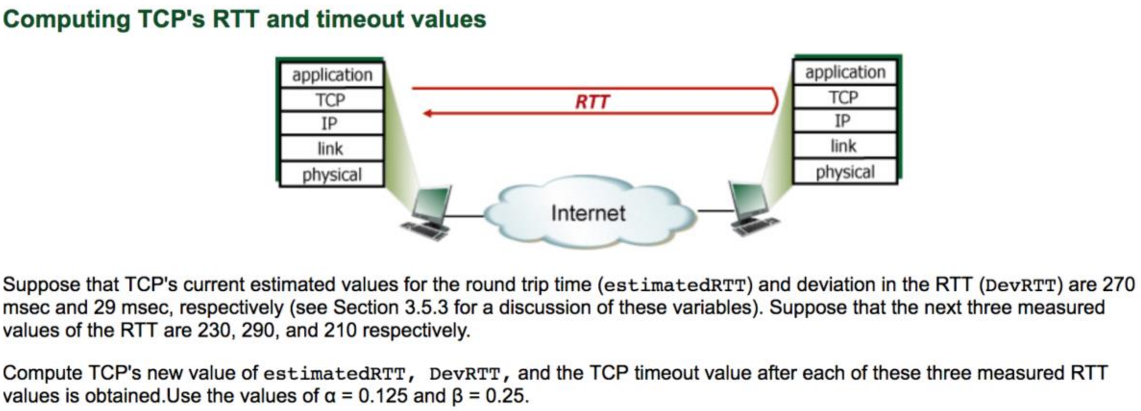


Consider the evolution of TCP's congestion window in the example above and answer the following questions. The initial value of cwnd is 1 and the initial value of ssthresh (shown as a red +) is 8.

- Give the times at which TCP is in slow start, congestion avoidance and fast recovery at the start of a time slot, when the flight of packets is sent.

- Give the times at which the first packet in the sent flight of packets is lost, and indicate whether that packet loss is detected via timeout, or by triple duplicate ACKs.
- Give the times at which the value of `ssthresh` changes, and give the new value of `ssthresh`.

4. (1 point)



Computing TCP's RTT and timeout values

Suppose that TCP's current estimated values for the round trip time (**`estimatedRTT`**) and deviation in the RTT (**`DevRTT`**) are 270 msec and 29 msec, respectively (see Section 3.5.3 for a discussion of these variables). Suppose that the next three measured values of the RTT are 230, 290, and 210 respectively.

Compute TCP's new value of **`estimatedRTT, DevRTT,`** and the TCP timeout value after each of these three measured RTT values is obtained.Use the values of α = 0.125 and β = 0.25.