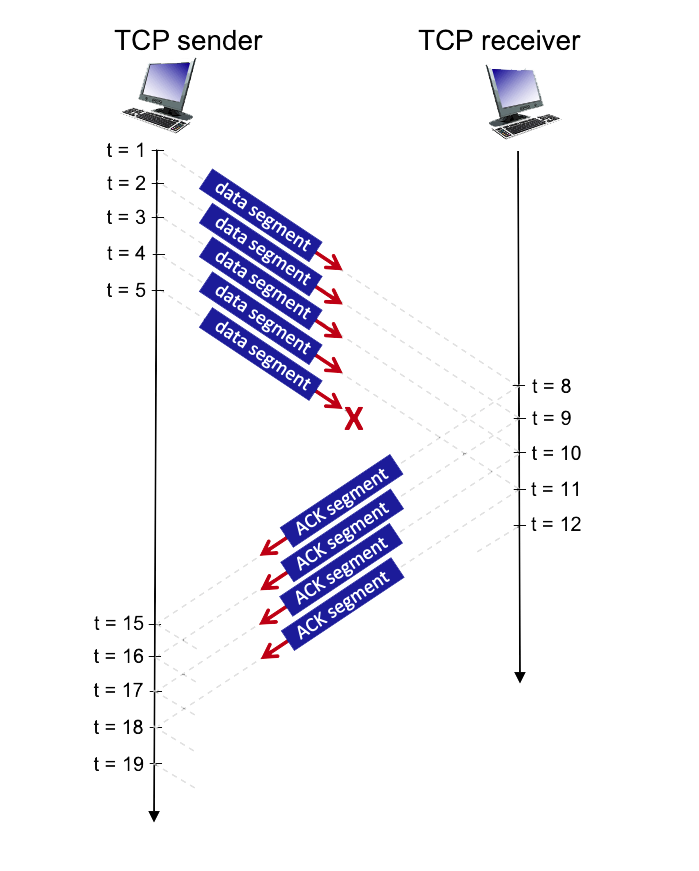
**1. TCP RETRANSMISSIONS (RELIABLE DATA TRANSMISSION WITH ACK LOSS)**

Consider the figure below in which a TCP sender and receiver communicate over a connection in which the segments can be lost. The TCP sender wants to send a total of 10 segments to the receiver and sends an initial window of 5 segments at t = 1, 2, 3, 4, and 5, respectively. Suppose the initial value of the sequence number is 164 and every segment sent to the receiver each contains 564 bytes. The delay between the sender and receiver is 7 time units, and so the first segment arrives at the receiver at t = 8, and an ACK for this segment arrives at t = 15. As shown in the figure, 1 of the 5 segments is lost between the sender and the receiver, but *none* of the ACKs are lost. Assume there are no timeouts and any out of order segments received are thrown out.

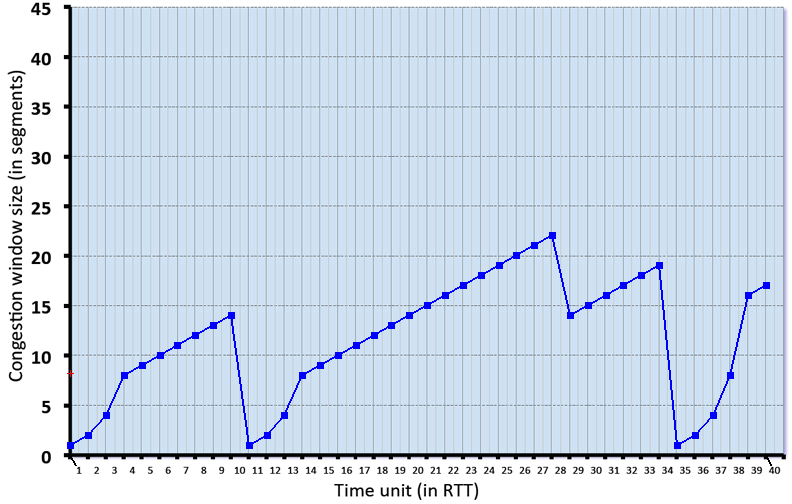


**QUESTION LIST**

1. What is the sequence number of the segment sent at t=1?  
2. What is the sequence number of the segment sent at t=2?  
3. What is the sequence number of the segment sent at t=3?  
4. What is the sequence number of the segment sent at t=4?  
5. What is the sequence number of the segment sent at t=5?  
6. What is the value of the ACK sent at t=8? (If segment lost, write 'x')  
7. What is the value of the ACK sent at t=9? (If segment lost, write 'x')  
8. What is the value of the ACK sent at t=10? (If segment lost, write 'x')  
9. What is the value of the ACK sent at t=11? (If segment lost, write 'x')  
10. What is the value of the ACK sent at t=12? (If segment lost, write 'x')  
11. What is the sequence number of the segment sent at t = 15? (If ACK never arrives, write 'x')  
12. What is the sequence number of the segment sent at t = 16? (If ACK never arrives, write 'x')  
13. What is the sequence number of the segment sent at t = 17? (If ACK never arrives, write 'x')  
14. What is the sequence number of the segment sent at t = 18? (If ACK never arrives, write 'x')  
15. What is the sequence number of the segment sent at t = 19? (If ACK never arrives, write 'x')

**2. SLOW START, CONGESTION AVOIDANCE, AND FAST RETRANSMIT**

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.



The initial value of *cwnd* is 1 and the initial value of *ssthresh* (shown as a red +) is 8.

**QUESTION LIST**

1. Give the times at which TCP is in slow start.   
2. Give the times at which TCP is in congestion avoidance.   
3. Give the times at which TCP is in fast recovery.  
4. Give the times at which packets are lost via timeout.   
5. Give the times at which packets are lost via *triple ACK*.   
6. Give the times at which the value of *ssthresh* changes