

Computer Network HW2

1. 18+2=20 points

(For simplicity, use sequence numbers 1, 2, ... to mark the segments in the problem)

P37. Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that five 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 five data segments to Host B, and the second segment (sent from A) is lost. In the end, all five 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 protocols are much longer than 5 RTT, then which protocol successfully delivers all five data segments in shortest time interval?

2. 5+10+5=20 points

P48. Consider that only a single TCP (Reno) connection uses one 10 Mbps 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?
- b. What is the average window size (in segments) and average throughput (in bps) of this TCP connection?
- c. How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?

3. 20 points

- P40. Consider Figure 3.61. 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.
- Identify the intervals of time when TCP slow start is operating.
 - Identify the intervals of time when TCP congestion avoidance is operating.
 - After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
 - After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

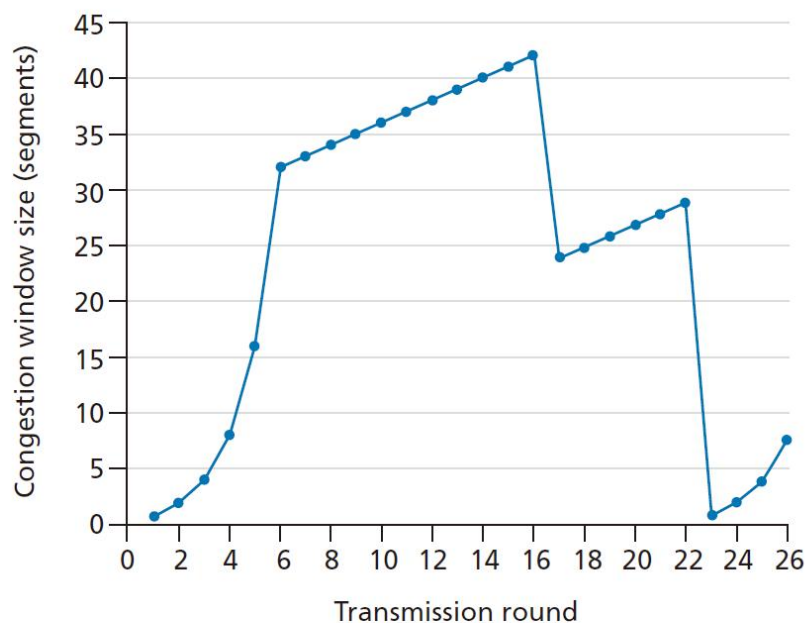


Figure 3.61 ♦ TCP window size as a function of time

- What is the initial value of `ssthresh` at the first transmission round?
- What is the value of `ssthresh` at the 18th transmission round?
- What is the value of `ssthresh` at the 24th transmission round?
- During what transmission round is the 70th segment sent?
- 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`?
- 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?
- 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?

4. 1*10=10 points

- P54. Consider a modification to TCP's congestion control algorithm. Instead of additive increase, we can use multiplicative increase. A TCP sender increases its window size by a small positive constant a ($0 < a < 1$) whenever it receives a valid ACK. Find the functional relationship between loss rate L and maximum congestion window W . Argue that for this modified TCP, regardless of TCP's average throughput, a TCP connection always spends the same amount of time to increase its congestion window size from $W/2$ to W .

5. 3*10 = 30 points

- P58. In this problem, we consider the delay introduced by the TCP slow-start phase. Consider a client and a Web server directly connected by one link of rate R . Suppose the client wants to retrieve an object whose size is exactly equal to $15 S$, where S is the maximum segment size (MSS). Denote the round-trip time between client and server as RTT (assumed to be constant). Ignoring protocol headers, determine the time to retrieve the object (including TCP connection establishment) when
- $4 S/R > S/R + RTT > 2S/R$
 - $S/R + RTT > 4 S/R$
 - $S/R > RTT$.