TELCOM 2310 Fall 2022
HW 5: Chapter 3

Due Wednesday November 2. Solutions must be typed and submitted on Canvas as Word or PDF documents. Please show your steps/reasoning (otherwise we cannot assign partial credit).

You may use your textbook and notes to complete this assignment, and may discuss *concepts* with other students or use online resources that help you better understand the concepts involved. **You are NOT permitted look at other students' solutions, or online solutions to substantially similar problems. Similarly, you are not permitted to share your solutions with other students, or post your solutions online. Ask the instructor if you have any questions about this policy.**

1. In lecture, we mentioned that for a reliable data transfer protocol using the Selective Repeat approach, the sequence number space must be at least twice the window size (assume the sender and receiver use the same window size, and the network does not re-order messages).

   (a) Explain in your own words why this is true, and give an example that shows why the sequence space cannot be smaller. Specifically, for your example, consider a window size of 4. In this case, we need at least 8 valid sequence numbers (e.g. 0-7). Give a specific scenario that shows where we could encounter a problem if the sequence space was less than 8 (i.e. explain what messages and acks are sent and received; it may be helpful to draw sender and receiver windows). (10pts)

   (b) Now consider Go-Back-N. Assuming again that our window size is 4 (i.e. N=4), what is the minimum number of valid sequence numbers that are needed? Explain in your own words why this is true, and give an example that shows why the sequence space cannot be smaller (as in part (a), show a specific scenario that does not work if the sequence space is smaller). In general, for a window of size N, how many distinct sequence numbers are needed? (15pts)

2. Consider TCP's algorithm for setting and dynamically adapting its timeout value, as described in lecture. Assume $\alpha = 0.125$ and $\beta = 0.25$. Assume that at some point in time the EstimatedRTT is 200ms and the DevRTT is 40ms. From this point on, all measured RTTs are exactly 20ms.

   (a) What is the timeout interval after 10 RTT measurements? (10pts)

   (b) How many such RTT measurements are needed before the timeout interval falls below 50ms? (15pts)

   Note: You will likely find it useful to write a small program or use a spreadsheet (particularly for part b). If you use this approach, please attach the code or spreadsheet to your solution.

3. Consider a TCP connection where the sender sends segments at a constant rate of one every 10ms, and the receiver sends ACKs back at the same rate without delay. Assume the roundtrip propagation delay is 300ms. A segment is lost, and upon receipt of the third duplicate ACK, the fast retransmit algorithm detects the loss and retransmits the lost segment. Assume that the lost segment is not the first data segment of the connection, and ignore transmission, processing, and queuing delays.

   If the sender must wait to receive an acknowledgment for the lost segment before sliding its window forward and transmitting new data:

   (a) How much total time does the sender lose compared with lossless transmission (i.e. how much longer does it wait to receive the ack and slide the window forward, compared to the case where there is no loss)?

   Hint: consider that the lost segment is initially sent at time $T = 0$. At what time would the sender receive the ACK for that segment in the no-loss case? At what time does that happen if the sender first needs to detect the loss, retransmit the segment, and *then* receive the ACK? (15pts)

   (b) Assuming a timeout of 400ms is set at the time the lost segment is originally transmitted, how much total time does the sender save by using fast retransmit, compared with waiting for the timeout to expire before retransmitting the segment (i.e. how much sooner is it able to slide the window forward than in the case where it waits for the timeout to occur)? (10pts)

4. Consider that only a single TCP (Reno) connection uses one 100Mbps 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 1460 bytes; the roundtrip propagation delay of this connection is 50msec; and this TCP connection is always in congestion avoidance phase (i.e. ignore slow start).

   (a) What is the maximum window size (in segments) that this TCP connection can achieve (round to the nearest whole segment if needed)? (10pts)

   (b) What is the average window size (in segments) and average throughput (in Mbps) of this TCP connection? (15pts)

5. (Bonus: 5pts) Recall that parallel HTTP connections are often used to improve load performance for webpages.

   (a) Given what we've learned about TCP, why are parallel connections not an ideal solution to HTTP performance problems (from a network perspective)?

   (b) Why are parallel HTTP connections commonly used anyway? (Recall our discussion from Chapter 2, and be as specific as you can)