Assignment (due Friday, 11pm of Week 9)

1. Throughout this exercise, answer the questions in the template corresponding to the paragraph numbers used here. Make any additional notes and record any questions you have in the **Additional Notes** and **Questions** sections at the end of the report template.

2. Working in teams of two, you will be running and modifying your TCP client and server code from Lab 3b, and they must be run on separate machines. You will need to locate the IP addresses of your machines for the TCP code as well as to add a filter to Wireshark. If your Lab3b is not correctly closing the connection, simplify the protocol that both the sender and receiver use to send data. For example, the sender could only send "X's" until the end, when it sends a 'Q'. The receiver could continually receive only one byte until it receives the 'Q', and then exit.

3. Fire up Wireshark on both machines. Since you may have quite a bit of network traffic that we are not interested in, create a capture filter for your partner's machine. I recommend something along the lines of `ip.host == 155.192.xxx.yyy`

4. With the capture running on both machines, invoke the server (tcp_receive) and client (tcp_send). Both machines should capture the same conversation. Locate the SYN, SYN/ACK, ACK three-way handshake, and the pair of FIN, ACKs at the end of the conversation. Also note the Data portion of the TCP packets and locate the actual messages exchanged. Note any discrepancies between the captures on the two machines.

5. In the segments containing data, note the win=XXXXX. This is the window advertisement and hence the amount of data the sender of that packet is able to receive.

6. Now, modify tcp_send to load the receiver and network:

    1. tcp_send: Modify to send large amounts of data. For example, instead of accepting input from the user, send a line of text with 10000 characters and send it 10 times in a loop. The code `message = 'X' * 10000` will result in a large string of Xs. Keep sending more data until you start to see the network and/or receiver slow down.

    2. As before, capture packets from this transaction. If you are not seeing any evidence of overwhelming the network or the receiver, add a sleep statement to the receiver, so that every time recv is called, it sleeps 100, 500, or 1000 ms. Run the capture and programs again. At this point you should see the window advertisement reach 0 and effectively halt transmission. Leave everything run for a few minute and observe the capture. You may need to exit with a Ctrl-C.

    3. Scroll through the capture and answer the questions in the template.

7. Be sure to fill in the required "Comments on lab or suggestions for improvement"

If you have time

- In your capture, try to locate any packets that were retransmitted. Try to characterize the cause of the retransmission (timeout, multiple ACKs) and describe below.

- Explore the sequence and acknowledgement numbers. Do they follow the pattern we discussed in class (in the afternoon section)

- Look at Wireshark's summary of the packet. From this alone, can you predict the byte-values for the source, destination, sequence, acknowledgement, and window fields of the packet? Can you predict the layout of the entire packet by putting in zeros for the fields you didn't predict so the byte alignment is correct?