EECS 325/425

Homework 3
(due Oct 15 before 11:59pm)
325: 29 pts + 5pts bonus; 425: 34pts
(Note two pages!)

1. Two black armies are stationed on opposite sides of a valley preparing to attack a white army in-between. The white army can defeat either of the black armies individually but will be defeated if both black armies attack at the same time. The black armies communicate over an unreliable channel (a messenger who might be intercepted by the enemy). Is there a protocol the black armies can use to *ensure* victory? (**3pts**: 1pt – correct answer; 2 pts – explanation)

2. Consider RDT 3.0. Give a scenario (sequence of events) and the corresponding time diagram leading to:
   a. Receiver is in state "wait for packet 1" but receives packet 0. **(2 pts)**
   b. Sender in state "Wait for ACK 0" but receives ACK 1. **(2 pts)**

3. In GBN protocol, if the receiver has sent ACK for packet number k, is it possible for it to send ACK for packet number (k-1) at some later point? Why or why not? **(2 pts)**

4. Consider GBN and SR with unlimited sequence numbers (i.e., the numbers never wrap around). Which one(s) will work correctly with the unreliable channel that may both lose and reorder packets? Justify your answer (**3 pts:** 1 for correct answer (0.5 for the answer regarding each protocol), 2 for the explanation).

5. (a) In GBN, assume sender cycles through only 4 distinct sequence numbers (1-2-3-4-1 etc.) for window size 4. Assume the network can lose packets but the network delay is zero ("negligible"). Give a specific packet exchange scenario where the receiver will deliver wrong data to the application. **(2 points)**

   (b) What range do we need in the GBN protocol to guarantee correct operation under these conditions (e.g., with possible packet loss but zero network delay) and why? (Hint: assume the receiver knows N.) **(2 pts)**.

6. Consider a sender and receiver communicating over a network connection that supports 300 packets/sec transmission rate. While the network normally delivers packets with negligible delay, it can have a maximum one-way packet delivery delay of 10 seconds. Let sender and receiver communicate using Selective Repeat with window size 120 and timeout value of 1 second. According to the discussion in class, the protocol uses 8 bits for sequence numbers, allowing 2^8=256 > 2*120 distinct sequence numbers before a wrap-around. However, as we talked in class, the above condition, while necessary, is not sufficient for protocol correctness.

   a. Give a specific scenario that leads to an incorrect execution of the protocol (i.e., that a sequence of packets sent will not be same as the sequence of packets delivered to the application at the receiver side) in the setup described **(3 pts)**
   b. How big the sequence number field has to be so that SR protocol always operates correctly under the above network parameters. **(2 pts)**

7. Read section 3.5.3 of 6^{th} edition of textbook to learn about RTT and timeout value (commonly abbreviated as RTO, for Retransmission Time Out) estimation. Note (this is not clear in the book) that: (i) after the very first RTT sample, the estimated RTT is set to this sample, (ii) the DevRTT is set to half that first RTT sample, and (iii) the new value of TimeoutInterval is computed as described in the book; (iv) When a new RTT sample is obtained, you first compute the new value

of DevRTT (using the value of EstimatedRTT that you had *before* this last sample) then compute the new value of EstimatedRTT, and then compute the new value of RTO (or TimeoutInternval as it is called in the book). Now answer the following questions:

a. Textbook, 6^{th} edition, Chapter 3, problem 32. Explain why this RTT estimation is called exponential moving average. **(a – 1 pt. b - 2pts, c – 2pts)**

b. Textbook, Ch. 3, problem 31 (only compute for just the first three RT samples) **(2 pts)**

8. In estimating round-trip time, TCP does not measure SampleRTT for retransmitted segments. Why? **(2 pts)**

425 students:

(I have not yet covered TCP handshake in class, but it's trivial – just three packets – so you can read it in the book).

We will use the trace you collected for HW 2.

Use packet exchange in the TCP handshake to compute the RTT between your client machine and all the servers you have accessed during the collection. In the output of your program, list the RTT for all servers (if you see multiple connections to the same server, i.e., the same IP address, take the average of the RTTs produced by all such connections). Paste this output into your lab report and comment on the range of RTT values you have observed. **(5 pts)**

Deliverables:

(1) The trace file in ASCII format (i.e., after processing by tcpdump).
(2) Your program(s), properly commented (readability will be considered in grading). Do not prepare a separate documentation (do not bother with class documentations produced with, e.g., Dr. Java etc.) Just insert sufficient comments within your source code.
(3) A lab report with results and explanations requested