CS 4310: Solutions #4

Due on Tuesday, Nov 30, 2010

Mina Guirguis

You may discuss this problem set with other students. However, you must write up your answers on your own. You must also write the names of other students you discussed each problem with. Each problem has a different weight. Please state any assumptions you are making in solving a given problem. Late assignments will not be accepted. Please start early, understand what each problem is asking and email me/come see me, if you have questions.

Problem 1

Consider the following TCP connection between a sender and a receiver. You can assume that this transfer took place over a network of 10 msec of propagation delay and 1 Mbps capacity, in each direction. You can assume that the receiver has piggybacked its request with the third segment of the 3-way handshake. You can assume that the request was of 50 bytes (without headers). Only focus on TCP/IP headers and ignore the link layer headers. Answer the following questions:

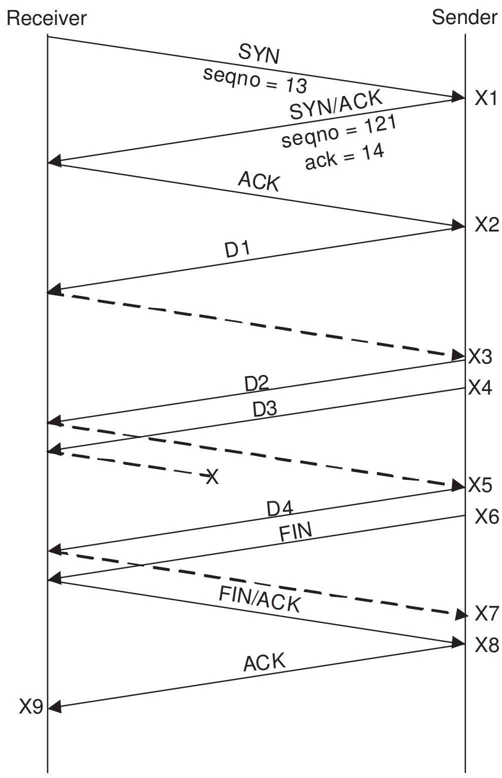

Figure 1: Time-line Diagram

(a) If segments D1, D2 D3, and D4, have the following sizes (in bytes with headers), 1000, 1000, 1000, and 500, respectively. What is the total size of the file being transferred?

(a) Accounting for the TCP/IP headers, the file size is $960 + 960 + 960 + 460 = 3340 bytes$.

(b) Complete the timing information (x1 through x9).

(b) The time it takes to transmit a packet of size p is $l + \frac{P}{B}$*, where l is the propagation delay and B is the link bandwidth. Thus, we have x1 = 10.32 msec. x2 = 31.36 msec. x3 = 59.68 msec. x4 = 67.68. x5 = 88*

msec. x6 = 92 msec. x7 = 112.32 msec. x8 = 112.64 msec. x9 = 122.96 msec.

(c) State the sequence numbers of D1, D2, D3 and D4.

(c) Sequence numbers are 122, 1082, 2042 and 3002, respectively.

(d) Can you tell whether the transmissions of D2 and D3 were triggered by slow start versus the normal AIMD?

(d) We can't really tell since in both cases we go from 1 to 2 packets. However, slow-start is used when a connection is started.

(e) For the sender and receiver specified, list the states (obtained from the TCP state transition diagram) they visited and how long they spent in each state (indicate the starting time and ending time in each state).

(e) Initially, the receiver is in CLOSED and the sender is in LISTEN. Let's consider the receiver first. The receiver moves to SYN_SENT at time 0. Then moves to ESTABLISHED at time 20.64. Then moves to CLOSE_WAIT at time 102.32. Then moves to LAST_ACK directly (since it sent its FIN with the ACK). At time 122.96 it goes to CLOSED. Let's consider the sender. The sender moves from LISTEN to SYN_RCVD at 10.32. Then moves to ESTABLISHED at time 31.36. Then moves to FIN_WAIT1 at time 92. Then moves to TIME_WAIT 112.64. After the timeout (120 seconds), it moves to CLOSED. All times are in milliseconds.

Problem 2

Consider a TCP connection starting with a congestion window (congwnd) size of 1 packet. Assume that the advertised window size is infinite (and hence we can ignore its effect). Furthermore, we can ignore the effect of extra queueing delay on our RTT calculations. Answer the following questions:

(a) How many RTTs would take this connection to transmit a 100-packet sized file, just by using TCP AIMD mechanism, when no packets are lost during the transfer? (*Assume slow-start is not used.*)

(a) We need to count how many packets would be transmitted in every RTT. In the first RTT, we send 1 packet. In the second RTT, we send 2 packets. In the third RTT, we send 3 packets, and so on. So to finish sending the file, the total number of packets $(1 + 2 + 3 + 4 + ... + n)$ *should be equal to 100. We need to find i. This summation equals to* $\frac{n(n+1)}{2}$*, which gives* $n = 14$.

(b) How many RTTs would take this TCP connection to transmit the same file, if slow start is used and the slow start congestion threshold was set to 32 packets? (*Again, you may assume no packets are lost.*)

(b) In this case, we are doing slow start till we reach 32 packets. So we have 1+2+4+8+16+32 for a total of 63 packets. The next RTT, we send 33 and then we send 4 and we are done. So it would take 8 RTTs.

Problem 3

Suppose that a scientist wants to establish the reliability of a given link by sending few packets (probe packets) over a link and observing how many of them make it to the other side. For example, if the scientist sends 10 probe packets and only 8 make it to the other side, the scientist is going to assume that the link has an 80% reliability. The scientist also wants to ensure that the probe packets are nicely spaced out when being sent to give accurate results. Answer the following questions:

(a) Is this method feasible with TCP? Explain why or why not.

(a) This method is not feasible with TCP. This is so because the congestion control of TCP would affect the rate since any packet loss would trigger a reduction in the congestion window.

(b) Is this method feasible with UDP? Explain why or why not.

(b) This method is feasible with UDP. This is so because one can keep a steady rate of UDP packets and observe the percentage of those that are lost.

Problem 4

Consider a TCP connection operating over a 1-Gbps network. Answer the following question:

(a) How long would it take the sequence numbers to wrap around completely, if the connection was able to utilize the full bandwidth continuously.

(a) The sequence number is 32 bits and counts in Bytes so the time taken to wrap around is $\frac{2^{32} * 8}{1Gbps} = 34$ *seconds.*

(b) If we added a 32-bit time-stamp field that gets incremented 1000 times during the wraparound time found above. How long would it take the timestamp to wrap around?

(b) It would take $\frac{2^{32}}{1000} \times 34$ *for the time-stamp to wrap around which is about 4.63 years.*

(c) If no packets are allowed to be buffered in this network, can TCP utilize the full bandwidth of the network? Explain why or why not. How long would it take the sequence numbers to wrap around then (assume no times-tamp is used)?

(c) TCP cannot utilize the full bandwidth of the network, since with no buffering, TCP would cut its congestion window in half as soon as a packet is lost. This means that the average TCP throughput is about 0.75 Gbps. This means that the sequence number would wrap around in 45 seconds.