EECS 489 Discussion 4

General Announcements

- Project 2 is out
  - Get started early, you could run into random issues
  - Due on Feb 24th
  - Team repos have been created

Q1 True or False

- With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.
- With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window
- The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.
- The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.

Q1 True or False

- With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window. T
- With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window T
- The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1. T
- The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1. T

Q2

Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 536 bytes.

- What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes
- For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

Q2

Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 536 bytes.

- What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes 2^32
- For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

(2^32 / 536) packets => (2^32 / 536)*66 bytes in headers = 528,857,934

=> bytes transferred = 2^32 + 528,857,934 => 38590.6 Mb => 249 seconds

Q3

Host A and B are communicating over a TCPconnection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?

If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

Q3

Host A and B are communicating over a TCPconnection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number? 207, 302, 80

If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number? 127

Q3

Host A and B are communicating over a TCPconnection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.

Q4

Suppose that the five measured SampleRTT values (see Section 3.5.3) are 106 ms, 120 ms, 140 ms, 90 ms, and 115 ms. Compute the Estimated RTT after each of these Sample RTT values is obtained, using a value of $\alpha = 0.125$ and assuming that the value of Estimated RTT was 100 ms just before the first of these five samples were obtained. Compute also the DevRTT after each sample is obtained, assuming a value of $\beta = 0.25$ and assuming the value of DevRTT was 5 ms just before the first of these five samples was obtained. Last, compute the TCP Timeout Interval after each of these samples is obtained.

Q4

Suppose that the five measured SampleRTT values (see Section 3.5.3) are 106 ms, 120 ms, 140 ms, 90 ms, and 115 ms. Compute the Estimated RTT after each of these Sample RTT values is obtained, using a value of $\alpha = 0.125$ and assuming that the value of Estimated RTT was 100 ms just before the first of these five samples were obtained. Compute also the DevRTT after each sample is obtained, assuming a value of $\beta = 0.25$ and assuming the value of DevRTT was 5 ms just before the first of these five samples was obtained. Last, compute the TCP Timeout Interval after each of these samples is obtained.

Ending Estimated RTT = 106.71 ms
Ending DevRTT = 12.28 ms
Timeout Interval = 158.23 ms