Elizabeth Dayton
Ead0044

COMP 4320
Introduction to Computer Networks
2021 Summer Mini-Semester II

Homework 3
Due in Canvas: 11:55pm July 29, 2021

Reference textbook: Computer Networking: A Top-Down Approach, 7th Edition, by James F. Kurose and Keith W. Ross, published by Addison-Wesley, 2017, ISBN 0-13-359414-9.

All homework assignments must be completed by each student individually. Any copying of someone else's work, or misrepresentation of other work as your own, will be grounds for failing this assignment or the course.

Penalty for late work is 20 points per day late.

There are 6 questions; make sure you answer all these questions.

1. In protocol `rdt3.0`, the ACK packets flowing from the receiver to the sender do not have sequence numbers (although they do have an ACK field that contains the sequence number of the packet they are acknowledging). Why is it that our ACK packets do not require sequence numbers?

   A duplicate ACK is obvious to the receiver in rdt 3.0, because when it received the original ACK, it transitioned to the next state. The duplicate ACK is not the one the sender needs and is thus ignored in rdt 3.0.

2. Suppose that the seven measured `SampleRTT` values (see Section 3.5.3) are 85 ms, 130 ms, 108 ms, 72 ms, 142 ms, 64 ms, and 153 ms. Compute the `EstimatedRTT` after each of these `SampleRTT` values is obtained, using a value of α = 0.2 and assuming that the value of `EstimatedRTT` was 110 ms just before the first of these seven samples were obtained. Compute also the `DevRTT` after each sample is obtained, assuming a value of β = 0.25 and assuming the value of `DevRTT` was 10 ms just before the first of these seven samples was obtained. Last, compute the TCP `TimeoutInterval` after each of these samples is obtained.

$$EstimatedRTT = (1 - \alpha) * EstimatedRTT + \alpha * SampleRTT$$
$$DevRTT = (1 - \beta) * DevRTT + \beta * |SampleRTT - EstimatedRTT|$$
$$TimeoutInterval = EstimatedRTT + 4 * DevRTT$$

   After obtaining 85 ms:

$$EstimatedRTT = (1 - 0.2) * 110 + 0.2 * 85 = 105\ ms$$
$$DevRTT = (1 - 0.25) * 10 + 0.25 * |85 - 110| = 13.75\ ms$$
$$TimeoutInterval = 105 + 4 * 13.75 = 160\ ms$$

After obtaining 130 ms:

$$EstimatedRTT = (1 - 0.2) * 105 + 0.2 * 130 = 110\ ms$$
$$DevRTT = (1 - 0.25) * 13.75 + 0.25 * |130 - 105| = 16.5625\ ms$$
$$TimeoutInterval = 110 + 4 * 16.5625 = 176.25\ ms$$

After obtaining 108 ms:

$$EstimatedRTT = (1 - 0.2) * 110 + 0.2 * 108 = 109.6\ ms$$
$$DevRTT = (1 - 0.25) * 16.5625 + 0.25 * |108 - 110| = 12.9219\ ms$$
$$TimeoutInterval = 109.6 + 4 * 12.9219 = 161.2875\ ms$$

After obtaining 72 ms:

$$EstimatedRTT = (1 - 0.2) * 109.6 + 0.2 * 72 = 102.08\ ms$$
$$DevRTT = (1 - 0.25) * 12.9219 + 0.25 * |72 - 109.6| = 19.0914\ ms$$
$$TimeoutInterval = 102.08 + 4 * 19.0914 = 178.4457\ ms$$

After obtaining 142 ms:

$$EstimatedRTT = (1 - 0.2) * 102.08 + 0.2 * 142 = 110.064\ ms$$
$$DevRTT = (1 - 0.25) * 19.0914 + 0.25 * |142 - 102.08| = 24.2986\ ms$$
$$TimeoutInterval = 110.064 + 4 * 24.2986 = 207.2582\ ms$$

After obtaining 64 ms:

$$EstimatedRTT = (1 - 0.2) * 110.064 + 0.2 * 64 = 100.8512\ ms$$
$$DevRTT = (1 - 0.25) * 24.2986 + 0.25 * |64 - 110.064| = 29.73995\ ms$$
$$TimeoutInterval = 100.8512 + 4 * 29.73995 = 219.881\ ms$$

After obtaining 153 ms:

$$EstimatedRTT = (1 - 0.2) * 100.8512 + 0.2 * 153 = 111.28096\ ms$$
$$DevRTT = (1 - 0.25) * 29.73995 + 0.25 * |153 - 100.8512| = 35.3422\ ms$$
$$TimeoutInterval = 111.28096 + 4 * 35.3422 = 252.64961\ ms$$

3. Host A and B are communicating over a TCP connection, and Host B has already received all bytes up through byte 256. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 90 and 140 bytes of data, respectively. In the first segment, the sequence number is 257, the port number is 3120, and the destination port number is 5470. Host B send an acknowledgement whenever it receives a segment from Host A.

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

    In the second segment from Host A to B, the sequence number is 257, the source port number is 3120, and the destination port number is 5470.

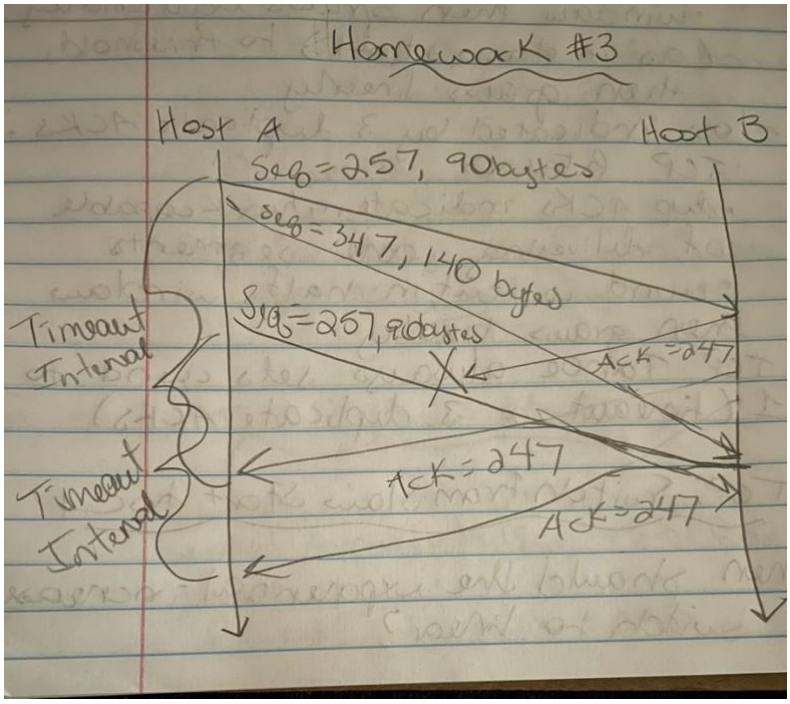
b. If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, what is the acknowledgement number?

If the second segment arrives before the first, the acknowledgement number is 257, indicating it is still waiting for bytes 257 and onward.

c. If the first segment arrives before the second segment, in the acknowledgement of the first arriving segment, what are the acknowledgement number, the source port number, and the destination port number?

If the first segment arrives before the second segment, the acknowledgement number is 257, the source port number is 5470, and the destination port number is 3120.

d. Suppose the two segments sent by A arrive in order at B. The first acknowledgement is lost and the second acknowledgement arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgements 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 acknowledgement that you add, provide the acknowledgement number.



4. Consider the GBN and SR protocols. Suppose the sequence number space is of size *X*. What is the largest allowable sender window that will avoid the occurrence of problems such as that in Figure 3.27 in the above textbook for *each* of these protocols?

   For selective repeat, the sequence number space X must be greater than or equal to the window size.

   For go-back-n, the sequence number space X must be greate than or equal to the window size + 1.

5. Consider a TCP connection has an initial Threshold of 32 kB and a Maximum Segment Size (MSS) of 6 kB. The receiver advertised window is 40 kB. Suppose all transmission attempts are successful except for a *timeout* at transmission *number 6* and a *triple duplicate ACK* received (for the same previously transmitted data) on the *number 12* transmission. The first transmission attempt is number 0. Find the size of the sender's *congestion window* for the first 18 transmission attempts (number 0-17) assuming the sender's TCP implementation is using the slow-start congestion control scheme.

   When it sends segment 0, the congestion window(cwnd) is one. Then the cwnd is doubled to two and segments 1 and 2 are sent with no error. Then the cwnd is doubled again to four and segments 3, 4, 5, and 6 are sent. Segment 6 has timeout, but we won't know this until the three duplicate ACKs are received, so cwnd is again doubled to eight, and segments 7, 8, 9, 10, 11, 12… and then we receive the triple duplicate ACKS. Segment 6 was not received, so we need to cut cwnd by half, back down to four, and retransmit. We now send segments 6, 7, 8, and 9. From, here it will keep increasing linearly every RTT, so now cwnd is 5, and we send segments 10, 11, 12, 13, 14, and 15. Assuming there is no triple duplicate ACKs received, we increase cwnd by one to six, and send remaining segments 16 and 17.

6. Compare GBN, SR and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 7 consecutive data segments and the corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host A sends 7 data segments to Host B, and the third segment (sent from A) is lost. In the end, all 7 data segments have been correctly received by Host B.

   a. How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.

      With go-back-n, Host A sends 12 segments in total. First it sends all the segments 1-7, and then must retransmit segments 3-7.
      The sequence numbers would be 0-11 for each of the 12 segments.
      Host B will send a total of 11 ACKs: 1 ACK for segment 1, 5 ACKs for segment 2, and then 1 ACK each for segments 3, 4, 5, 6, and 7.

With selective repeat, Host A sends 8 segments. First it sends all 1-7 segments, and then later retransmits segment 3.
The sequence numbers would be 0-8.
Host B will send a total of 7 ACKs: first one each for segments 1 and 2 and 4-7, and then 1 ACK later for the retransmitted segment 3.

With TCP, Host A will send 6 segments. First it sends all 1-7 segments and then retransmits segment 3.
The sequence numbers are 3 and 8.
Host B will send a total of 6 ACKs: 5 of them will have sequence number 3, and then 1 will have sequence number 8.

b. If the timeout values for all three protocols are much longer than 7 RTT, then which protocol successfully delivers all 7 data segments in the shortest time interval?

TCP will deliver all 7 data segments in the shortest time interval. This is because TCP uses fast retransmit and doesn't wait for the timeout.