COMP 4320
Introduction to Computer Networks
2022 Summer Mini-Semester II

Homework 3
Due in Canvas: 11:55pm July 31, 2022

Reference textbook: Computer Networking: A Top-Down Approach, 8th Edition, by James F. Kurose and Keith W. Ross, published by Pearson Education, Inc., 2021, ISBN 9780135928615.

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?
   Regular packets require sequence numbers to handle duplicates. However, ACK packets do not require sequence numbers because duplicates are easily detectible due to state changes. Looking at the FSM for rdt3.0, once the first ACK packet that the sender is waiting for arrives it immediately changes states and therefore any duplicates that arrive can be detected since they will no longer be the ACK packets that the sender is looking for.

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 $\alpha = 0.2$ 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 $\beta = 0.25$ 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 – α) * EstimatedRTT + α * SampleRTT
   DevRTT = (1 – β) * DevRTT + β * |SampleRTT – EstimatedRTT|
   TimeoutInterval = EstimatedRTT + 4 * DevRTT

   1st Sample:
   EstimatedRTT = (1 – 0.2) * 110 + 0.2 * 85 = 105 ms
   DevRTT = (1 – 0.25) * 10 + 0.25 * |85 – 105| = 12.5 ms
   TimeoutInterval = 105 + 4 * 12.5 = 155 ms

   2nd Sample:
   EstimatedRTT = (1 – 0.2) * 105 + 0.2 * 130 = 110 ms

DevRTT = (1 – 0.25) * 12.5 + 0.25 * |130 – 110| = 14.375 ms
TimeoutInterval = 110 + 4 * 14.375 = 167.5 ms

3rd Sample:
EstimatedRTT = (1 – 0.2) * 110 + 0.2 * 108 = 109.6 ms
DevRTT = (1 – 0.25) * 14.375 + 0.25 * |108 – 109.6| = 11.181 ms
TimeoutInterval = 109.6 + 4 * 11.181 = 154.324 ms

4th Sample:
EstimatedRTT = (1 – 0.2) * 109.6 + 0.2 * 72 = 102.08 ms
DevRTT = (1 – 0.25) * 11.181 + 0.25 * |72 – 102.08| = 15.906 ms
TimeoutInterval = 102.08 ms + 4 * 15.906 = 165.704 ms

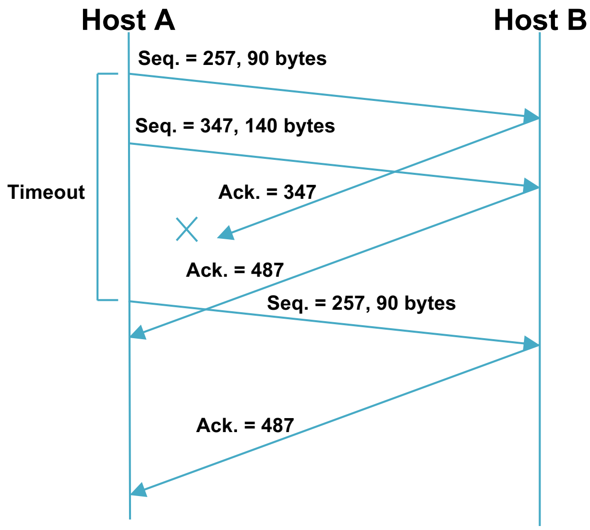
5th Sample:
EstimatedRTT = (1 – 0.2) * 102.08 + 0.2 * 142 = 110.064 ms
DevRTT = (1 – 0.25) * 15.906 + 0.25 * |142 – 110.064| = 19.914 ms
TimeoutInterval = 110.064 + 4 * 19.914 = 189.72 ms

6th Sample:
EstimatedRTT = (1 – 0.2) * 110.064 + 0.2 * 64 = 100.851 ms
DevRTT = (1 – 0.25) * 19.914 + 0.25 * |64 – 100.851| = 24.148 ms
TimeoutInterval = 100.851 + 4 * 24.148 = 197.443 ms

7th Sample:
EstimatedRTT = (1 – 0.2) * 100.851 + 0.2 * 153 = 111.281 ms
DevRTT = (1 – 0.25) * 24.148 + 0.25 * |153 – 111.281| = 28.541 ms
TimeoutInterval = 111.281 + 4 * 28.541 = 225.444 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?
    Sequence number = 347
    Source port number = 3120
    Destination port number = 5470
    b. If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, what is the acknowledgement number?
    Acknowledgement number = 257
    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?
    Acknowledgement number = 347
    Source port number = 5470
    Destination port number = 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?
If sequence number size = X
Let window size = N
For SR, the window size can be at most half the sequence number size (the sequence number size must be twice as large as the window size).
$X \geq 2N$
$X/2 \geq N$
For GBN, the window size can be at most one less than the sequence number size.
$X \geq N + 1$ **or**
$X - 1 \geq N$

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.

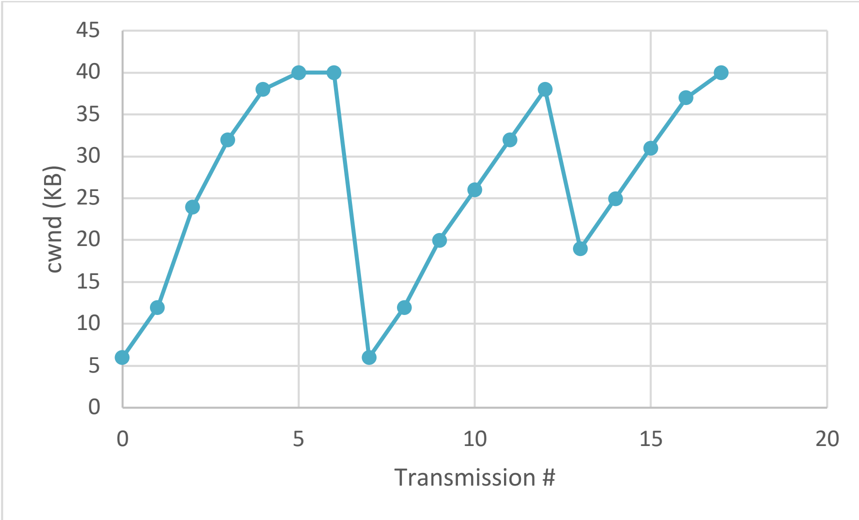


Chart for graph:

Transmission #	**cwnd (KB)**	Threshold (KB)
0	**6**	32
1	**12**	32
2	**24**	32
3	**32**	32
4	**38**	32
5	**40**	32
6	**40**	32
7	**6**	20
8	**12**	20
9	**20**	20
10	**26**	20
11	**32**	20
12	**38**	20
13	**19**	19
14	**25**	19
15	**31**	19
16	**37**	19
17	**40**	19

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.
    Go-Back-N:
        Host A sends 12 segments (1, 2, 3, 4, 5, 6, 7 and retransmit 3, 4, 5, 6, 7).
        Host B sends 11 ACKS. 1 ACK with sequence number 1, 5 ACKS with sequence number 2, and 1 ACK each for

sequence numbers 3, 4, 5, 6, 7.

Selective Repeat:

Host A sends 8 segments (1, 2, 3, 4, 5, 6, 7 and retransmit 3). Host B sends 7 ACKS. There is 1 ACK each for sequence numbers 1, 2, 4, 5, 6, 7 and 1 ACK with sequence number 3 for the retransmitted packet.

TCP:

Host A sends 8 segments (1, 2, 3, 4, 5, 6, 7 and retransmit 3). Host B sends 7 ACKS. There is 1 ACK with sequence number 2, 5 ACKS with sequence number 3 and 1 ACK with 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?

Since TCP uses fast retransmit (detecting duplicate ACKs and retransmitting accordingly) to avoid waiting for a timeout it will successfully deliver the segments the fastest.