Problem 1

Answer True or False to the following questions and briefly justify your answer:

(a) With the Selective Repeat protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

(b) With Go-Back-N, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

(c) The Stop&Wait protocol is the same as the SR protocol with a sender and receiver window size of 1.

(d) Selective Repeat can buffer out-of-order-delivered packets, while GBN cannot. Therefore, SR saves network communication cost (by transmitting less) at the cost of additional memory.

Write your solution to Problem 1 in this box

a) True, this is possible. Suppose we have a window size of five packets. The following could occur in chronological order. Five packets are sent, are ACKed by the receiver, and timeout at the sender. These five packets are resent by sender. Then, the original packet ACKs are received by the sender and the window is shifted. When the second group of packets are ACKed and received, the sender will receive ACKs for packets outside of your current window.

b) True, one can create a situation that is almost identical to that described above except for the GBN protocol.

c) True, if the SR protocol has a window size of 1, the protocol becomes stop-and-wait because the window doesn't move until we have sent that one packet and received its corresponding ACK.

d) Technically, GBN could buffer out-of-order packets, but it doesn't have to since it will always resend all packets regardless. Therefore, if we don't buffer the out-of order packets we do use more memory and SR only resends packets it needs to, saving communication cost. So this is true.

Problem 2

Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 226. 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 227, the source port number is 30002, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A. Fill in the blanks for questions (a) – (c) directly; work out the diagram in the box for question (d).

(a) In the second segment sent from Host A to B, the sequence number is 307, source port number is 30002, and destination port number is 80.

(b) If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, the ACK number is 307, the source port number is 80, and the destination port number is 30002.

(c) If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, the ACK number is 227.

(d) 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 in the box below, showing these segments and all other segments and acknowledgment sent. Assume no additional packet loss. For each segment in your diagram, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the ACK number.

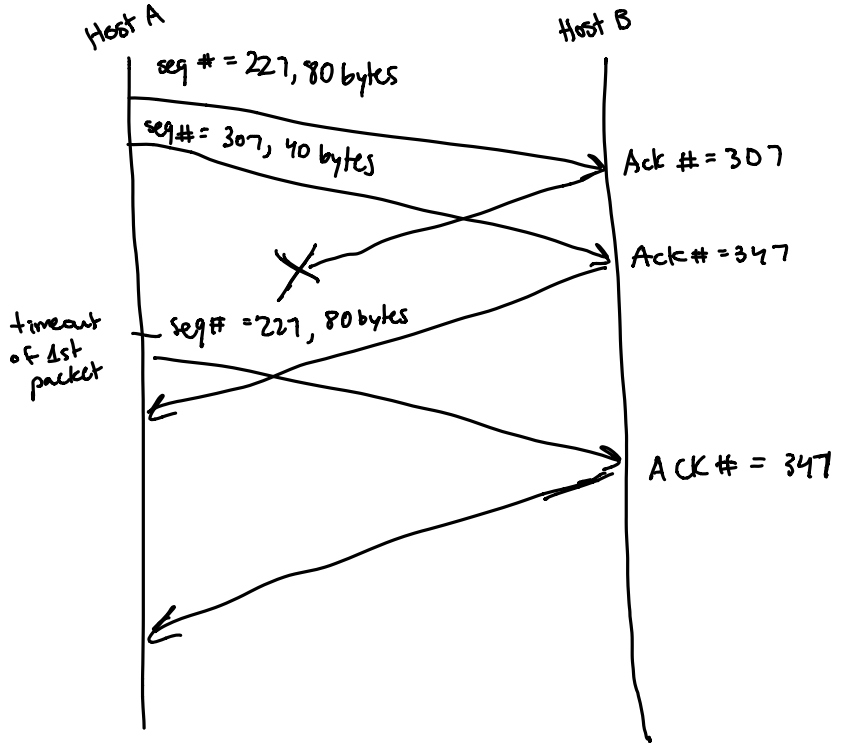
Write your solution to Problem 2 in this box

a) 307, 30002, 80

b) 307, 80, 30002

c) 227

d)



Problem 3

In Fast Retransmit algorithm, we saw TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first or second duplicate ACKs for a segment received?

Write your solution to Problem 3 in this box

This design decision was likely made because it was found to be the best tradeoff between a quick retransmission if a packet is lost, and not resending data if the packets are only received out of order.

In many TCP implementations, packets that are received out of order are buffered and would produce duplicate ACKs. If the threshold for retransmission was set too low, then we would resend a lot of these packets that were received out of order but don't necessarily need to be resent because we have them buffered.

As stated above, the choice of 3 was likely a tradeoff between providing fast retransmission and not unecessarily retransmitting.

Problem 4

Suppose that three measured SampleRTT values are 106 ms, 120 ms, and 140 ms. Compute the EstimatedRTT after each of these SampleRTT values is obtained, assuming that the value of EstimatedRTT was 100 ms just before the first of these three samples were obtained. Compute also the DevRTT after each sample is obtained, assuming the value of DevRTT was 5 ms just before the first of these three samples was obtained. Last, compute the TCP TimeoutInterval after each of these samples is obtained.

Write your solution to Problem 4 in this box

EstimatedRTT = $(1 - \alpha) *$ EstimatedRTT + $\alpha *$ sample RTT

$\alpha = 0.125$

① $(1 - 0.125) * 100 + (0.125 * 106) =$

$87.5 + 13.25 = 100.75$ ms

② $(.875 * 100.75) + (0.125 * 120) = 103.16$ ms

③ $(.875 * 103.16) + (0.125 * 140) = 107.76$ ms

DevRTT = $(1-\beta) *$ DevRTT + $\beta *$ |Sample RTT − EstimatedRTT|

$\beta = 0.25$

① $(.75 * 5) + .25 * (6) = 5.25$ ms

② $(.75 * 5.25) + .25 * (19.25) = 8.75$ ms

③ $(.75 * 8.75) + .25 * (36.84) = 15.77$ ms

TCP Timeout Interval = EstimatedRTT + 4* DevRTT

① $100.75 + 4(5.25) = 121.75$ ms

② $103.16 + 4(8.75) = 138.16$ ms

③ $107.76 + 4(15.77) = 170.84$ ms

Problem 5

Compare Go-Back-N, Selective Repeat, and TCP (no delayed ACK). Assume that timeout values for all three protocols are sufficiently long, such that 5 consecutive data segments and their 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 5 data segments to Host B, and the 2nd segment (sent from A) is lost. In the end, all 5 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.

(b) If the timeout values for all three protocols are much longer than 5RTT, then which protocol successfully delivers all five data segments in shortest time interval?

Write your solution to Problem 5 in this box

a) GBN: A sends data segments w/ sequence #'s 1, 2, 3, 4, 5 initially and resends segments 2, 3, 4, 5 after timeout. This is a total of 9 segments sent by A

B sends 8 ACKs in total. When packets 1, 3, 4, 5 are received, 4 ACKs are sent w/ seq # 1. Then when packets 2, 3, 4, 5 are received, 4 ACKs are sent w/ seq #'s 2, 3, 4, 5.

Selective Repeat: A initially sends data segments w/ sequence #'s 1, 2, 3, 4, 5 and later resends segment 2 once timeout has occurred. A sends 6 packets in total.

B initially sends ACKs for seq #'s 1, 3, 4, 5 and later 2 is sent once B receives it. This is 5 ACK packets in total.

TCP: A sends all data segments w/ seq #'s, 1, 2, 3, 4, 5 initially and then later resends data segment 2. This is a total of 6 data segments.

B sends 4 ACKs with seq #'s 2 that correspond with receiving data segments 1, 3, 4, 5. Once data segment 2 is successfully received, an ACK w/ seq # 6 is sent because TCP sends ACKs with a seq # corresponding to the next byte we expect. B sends 5 ACKs in total.

b) TCP will deliver all five data segments in the shortest time interval because TCP uses fast retransmission and doesn't wait for timeout to occur like the other two protocols.