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. Say the sender sends packets 1, 2, 3 at t_0. At t_1 ($t_1 > t_0$), the receiver receives these packets and sends ACK1, ACK2, & ACK 3. However, before the sender can receive these ACKs, the sender timeout and resends packets 1, 2, 3. The receiver will attempt to ACK these packets again at t_2 ($t_2 > t_1$). However, at t_3, the ACK messages from t_1 are finally received. The sender moves its window to 4, 5, 6. Now, at t_4 ($t_4 > t_3 > t_2$), the sender receives the ACK messages for packets 1, 2, 3, outside of its window.

b. True. Similarly to (a), a sender can send a packet 1, the receiver can receive and ACK it, but the sender can timeout before that ACK is received, sending packet 1 again. The sender can then receive the initial ACK and move its window to packet 2. Then, the sender can receive the ACK for the duplicate packet 1 after 1 has left its window.

c. True. An SR protocol with sender and receiver windows of size 1 functions the same as a Stop&Wait protocol. No out-of-order packets can be buffered and the ACK is an ACK for a single packet. The window only accepts another new packet once the old packet has been ACK-ed, like in Stop&Wait.

d. True. SR can request only necessary packets, while GBN requests an entire window. This saves communication costs.

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 ______, source port number is ______, and destination port number is ______.

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

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

(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.

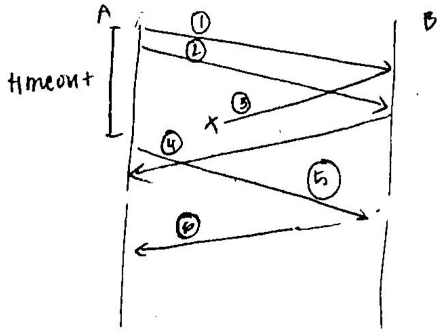
Write your solution to Problem 2 in this box

a. the sequenu number : 226 + 80 = 307.
the soure port number is: 30002.
the destination port number is 80.

b. the ACK number is: 307.
the source port number is: 80
the destination port number is : 30002.

c. The Ack number is 227; the sender is still waiting for an acknowledgment for the first segment.

d.



① seq # : 227 , 80 bytes
② seq # : 307 , 40 bytes
③ ACK : 307
④ : ACK : 347
⑤ : seq · # : 227, 80 bytes
⑥ : ACK : 347

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

TCP designers may have chosen to perform a fast retransmit after three duplicate ACKs rather than one or two ACKs because typically, one or two duplicate ACKs does not indicate packet loss but rather, out of order delivery. Sending a retransmission, in this case, would be wasteful. Moreover, if the packets are simply out of order, there would be needless transmission of redundant packets.

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

SampleRTT : 106ms

EstimatedRTT = $(1-0.125)(100ms) + (0.125)(106\ ms) = \boxed{100.75ms}$

DevRTT = $(1-0.25)(5ms) + 0.25\,|106ms - 100.75ms| = \boxed{5.0625ms}$

TimeoutInterval = $100.75 + 4(5.0625) = \boxed{121\ ms}$

SampleRTT: 120ms

EstimatedRTT = $(1-0.125)(100.75ms) + (0.125)(120ms) = \boxed{103.15625ms}$

DevRTT = $(1-0.25)(5.0625ms) + 0.25\,|120ms - 103.15625\ ms|$

$= \boxed{8.00\ ms}$

TimeoutInterval = $103.15625 + 4(8) = \boxed{135.15625\ ms}$

SampleRTT: 140 ms

EstimatedRTT = $(1-0.125)(103.15625ms) + (0.125)(140ms) =$

$= \boxed{107.7617188\ ms}$

DevRTT = $(1-0.25)(8ms) + 0.25(|107.7617188 - 140|)$

$= \boxed{14.0595703\ ms}$

Timeout Interval = $107.7617188\ ms + 4(14.0595703ms)$

$= \boxed{164\ ms}$

Problem 5

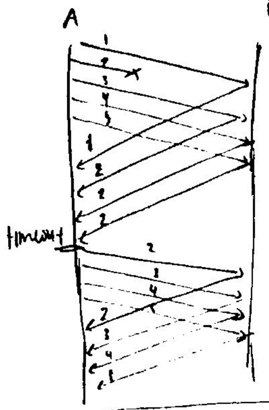
Compare Go-Back-N, Selective Repeat, and TCP (no delayed ACK). Assume that timeout values for all three protocols are sufficiently long, such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (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?

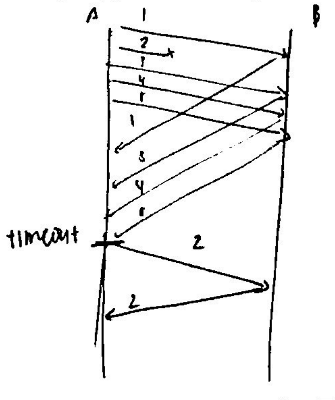
Write your solution to Problem 5 in this box

a. Go-Back-N:



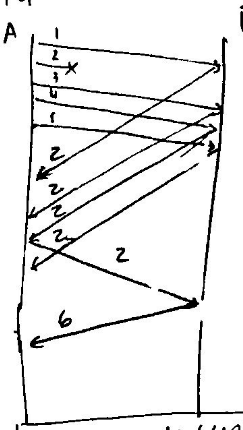
9 segments sent
8 ACKs sent

b Selective Repeat:



6 segments sent
5 ACKs sent

TCP:



6 segments sent
5 ACKs sent

b. TCP will deliver the segments in the shortest time interval due to fast retransmission, which causes the segment to be retransmitted after the third duplicate ACK number.