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
A simple example would be a window of size 2
sender sends packet 1, 2, and receiver sends back Ack_1, ACK_2.
The sender times out Ack1 & ACK2, and then retransmit
Packet 1 & 2. Receiver receives the duplicate and resends
Ack 1-1, Ack 1-2. After this the sender receives Ack_1 & Ack_2
due to delay, and the window moves on to packet 3 & 4.
Then Ack1-1 & Ack1-2 arrives, and packet 1 & 2 are out of
the window

(b) True
Same example as question (a)

(c) True
If the window size is 1, there would be no out-of-order
packets (within the window). Cumulative Acks are just ordinary
Acks. Thus, in this situation Stop& Wait is the same as SR
protocol.

(d) True Instead of discarding the out-of-order packets, as GBN
does, SR protocol holds them in a buffer (cost of additional
memory), so that the send only needs to retransmit the lost
packet, rather than everything after it.

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).

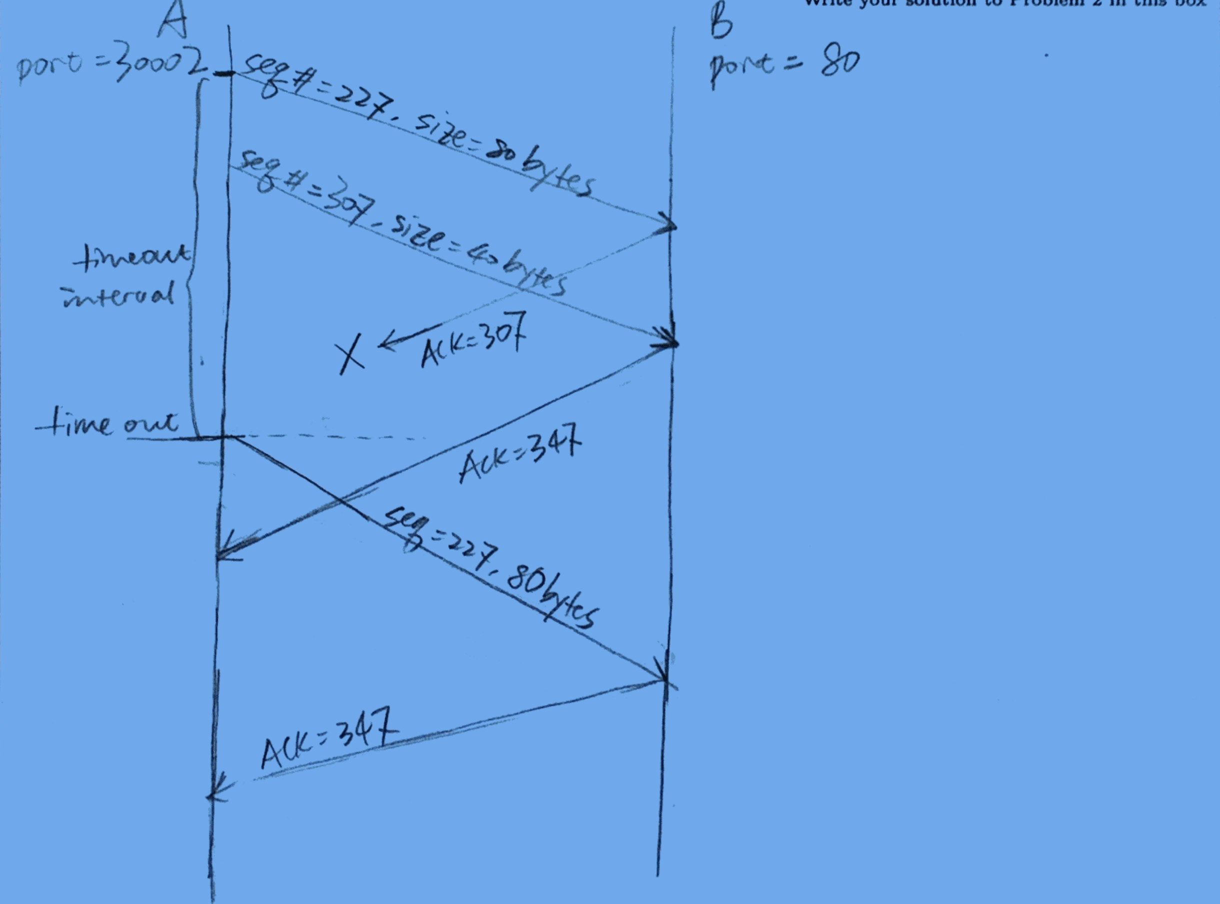
(a) In the second segment sent from Host A to B, the sequence number is 307, source port number is 30002, and destination port number is 80.

(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



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

Whenever the receiver receives out-of-order packet, it issues duplicate ACKs. Therefore, performing retransmittion after only the first or second duplicate ACK would cause too many unecessary redundant packets in the network. This would make network congestion even worse, while at the same time, the retransmittions are mostly triggered but out-of-order packets, rather than actual packet loss.

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

First:

EstimatedRTT = 0.125 × SampleRTT + (1−0.125) × EstimatedRTT
= 0.125 × 106 + 0.875 × 100
= 100.75 ms

DevRTT = 0.25 × |SampleRTT − EstimatedRTT| + (1 − 0.25) × DevRTT
= 0.25 × 5.25 + 0.75 × 5
= 5.0625 ms

TCPTimeoutInterval = EstimatedRTT + 4 × DevRTT
= 100.75 + 4 × 5.0625
= 121 ms

Second:

EstimatedRTT = 0.125 × 120 + 0.875 × 100.75
= 103.15625 ms

DevRTT = 0.25 × 16.84 + 0.75 × 5.0625
= 8 ms

TCPTimeoutInterval = 103.15625 + 4×8
= 135.16 ms

Third:

EstimatedRTT = 0.125 × 140 + 0.875 × 103.15625
= 107.75 ms

DevRTT = 0.25 × 32.25 + 0.75 × 8
= 14.06 ms

TCPTimeoutInterval = 107.75 + 4×14.06
= 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) GBN:

lost

9 segments sent (from A to B): 1 2 3 4 5 2 3 4 5
8 ACK with seq # (from B to A): 1 1 1 1 2 3 4 5

SR:

6 segments sent (from A to B): 1 2 3 4 5 2
5 Ack (from B to A): 1 3 4 5 2

TCP

6 segments sent (from A to B): 1 2 3 4 5 2
5 ACK (from B to A): 2 2 2 2 6

(b) TCP

Because TCP uses fast retransmit.
i.e. TCP retransmits after three duplicate ACKs
while GBN & SR protocols don't retransmit until timeout.