Problem 1

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

(a) The Stop-and-Wait protocol is the same as the Selective Repeat protocol with a sender and receiver window size of 1.

(b) 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.

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

(d) Selective Repeat can buffer out-of-order delivered packets, while Go-Back-N **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. If the sending and receiver window size is 1, then SR (and GBN) will be identical to Stop and Wait protocol.

b) True. It is possible if the premature timeout is triggered. A scenario may happens as below: The delay is too high, the sender retransmits the packets upon timeout, later the sender receives the ACKs for the original packets. It slide the sender window, move on. However, the ACKs for duplicated packets will be outside of its current window.

c) False. Same reason and scenario in (b), thus it should be possible.

d) True. Selective Repeat (SR) can buffer out-of-order delivered packets, while GBN cannot. When the loss rate is high, it will cost too much resources if we choose resending the whole window of packets. Therefore, SR saves network communication cost.

Problem 2

Consider a reliable data transfer protocol that uses only negative acknowledgments. Suppose the sender sends data only infrequently. Would a NAK-only protocol be preferable to a protocol that uses ACKs? Why? Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

Write your solution to Problem 2 in this box

In the first case, a NAK-only protocol would not be preferable to a protocol that uses ACKs.
With a NAK-only protocol, a lost packet will be detected only when the receiver correctly receive the subsequent packet. Since the sender sends data only infrequently, which means the error recovery time could be very long.

In the second case, a NAK-only protocol could be preferable to a protocol that uses ACKs. A lost packet could be soon detected in this scenario since the sender needs to send a lot of data which would frequently transmit data. Furthermore, the data loss rate is less in this case, So the receiver will rarely send NAKs. And compare to a protocol that use ACKs whose receiver has to send acknowledgement for every packet the receiver receive, NAK-only protocol does not need the receiver replies each time, the receiver sends less number of acknowledgements so that it could save costs.

Problem 3

One of the three functions of a sliding window scheme is the orderly delivery of packets which arrive out of sequence. In Go-back-N, the receiver drops packets which arrives out of order. Assume the receiver sends an ACK for every packet it receives.

(a) In Go-back-N, what is the required buffer size (receiver's window size, RWS) at the receiver if sender's window size (SWS) = 15? What the answer will be in Selective Repeat?

(b) In sliding window with SWS = RWS = 10, the minimum required `SeqNumSize` (the number of available sequence numbers) is 20. Calculate the minimum required `SeqNumSize` for

(i) a sliding window scheme with SWS = 8 and RWS = 6

(ii) a Go-back-N scheme with SWS = 9

Write your solution to Problem 3 in this box

a)

In GBN, at the receiver side, the buffer size (RWS) is always 1. It can only hold the next incoming packet.

In SR, at the receiver side, the buffer size is same as the window size in sender. Thus, the buffer size is 15 (or ≤ 15).

b) According to the lecture, in SR, the relationship between seq# size and window size should be seq# size = SWS + RWS. In GBN, the minimum sequence number required is N+1 (N = SWS) .

(i) Minimum required SeqNumSize = 8+6 = 14
(ii) Minimum required SeqNumSize = 9+1 = 10

6th

A $\xrightarrow{10}$ B

Problem 4

Compare Go-Back-N, Selective Repeat, and TCP (no delayed ACK). Assume that timeout values for all three protocols are sufficiently long, such that 10 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 10 data segments to Host B, and the 6th segment (sent from A) is lost. In the end, all 10 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 very long (e.g., longer than several RTTs), then which protocol successfully delivers all 10 data segments in shortest time interval?

Write your solution to Problem 4 in this box

a) Assume the sequence number starts from 1.
GBN:
Host A sends **15** segments in total: Initially sent 1-10, later resend segments 6-10.
Host B sends **14** ACKs in total : 9 ACKs with sequence number 1 2 3 4 5 5 5 5 5, later 5 ACKs with sequence number 6 7 8 9 10.

SR:
Host A sends **11** segments in total: initially sent 10 segments 1-10, later only resent the 6th segment 6.
Host B sends **10** ACKs in total: 9 ACKs initially with sequence number 1 2 3 4 5 7 8 9 10, later sent 1 ACKs with segment number 6.

TCP:
Host A sends **11** segments in total: initially sent 10 segments 1-10, later only resend the 6th segment 6. (Same as SR here)
Host B sends **10** ACKs in total: 9 ACKs initially: 2 3 4 5 6 6 6 6 6 (TCP always send an ACK with expected sequence number). And then sends one ACK with sequence number 11.

b) **TCP** protocol successfully delivers all five data segments in shortest time interval.
Reason: TCP uses fast retransmit without waiting until time out. Furthermore, unlike in GBN, TCP only retransmits the missing packet.

Problem 5

Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 326. 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 327, the source port number is 40320, 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 407, source port number is 40320, 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 407, the source port number is 80, and the destination port number is 40320.

(c) If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, the ACK number is 327.

(d) Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after A's timeout intervals for both the first and the second packets. Draw a timing diagram in the box below, showing these segments and acknowledgments until A receives all the acknowledgments of re-transmitted packets. 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 5 in this box