Computer Communications and Networks (COMN) 2021/22, Semester 2

Assignment 2 Results Sheet

Forename and Surname:	Hao Zhou
Matriculation Number:	S1862323

Question 1 – Number of retransmissions and throughput with different retransmission timeout values with stop-and-wait protocol. For each value of retransmission timeout, run the experiments for **5 times** and write down **average number of retransmissions** and **average throughput**.

Retransmission timeout (ms)	Average number of re-transmissions	Average throughput (Kilobytes per second)
5	986	65.8
10	611	45.6
15	122	75.1
20	110	68.8
25	102	67.2
30	101	62.8
40	124	50.7
50	112	46.8
75	112	42.3
100	88	36.5

Question 2 – Discuss the impact of retransmission timeout value on the number of retransmissions and throughput. Indicate the optimal timeout value from a communication efficiency viewpoint (i.e., the timeout that minimizes the number of retransmissions while ensuring a high throughput).

Increasing the retransmission timeout will reduce the average number of re-transmissions and the average throughput. As the retransmission timeout increases, the average number of retransmissions decreases significantly until 15ms. After that, the average number of re-transmissions will gradually decrease. The average throughput shows a similar trend. The average throughput peaks at 75 Kilobytes at 15ms of retransmission timeout. then, the average throughput gradually decreases to 36.8KB/s.

This is because when the retransmission timeout is small, the socket does not wait long enough for the ACK to arrive. As a result, the socket has to retransmit the packet frequently. So, the number of re-

transmissions is very high before 15ms. As the timeout increase, the average number of re-transmissions will decrease, because the socket has enough time to receive ACK and then retransmit the lost packets. Therefore, the throughput has to be smaller because socket have to wait for timeout when the packet is loss.

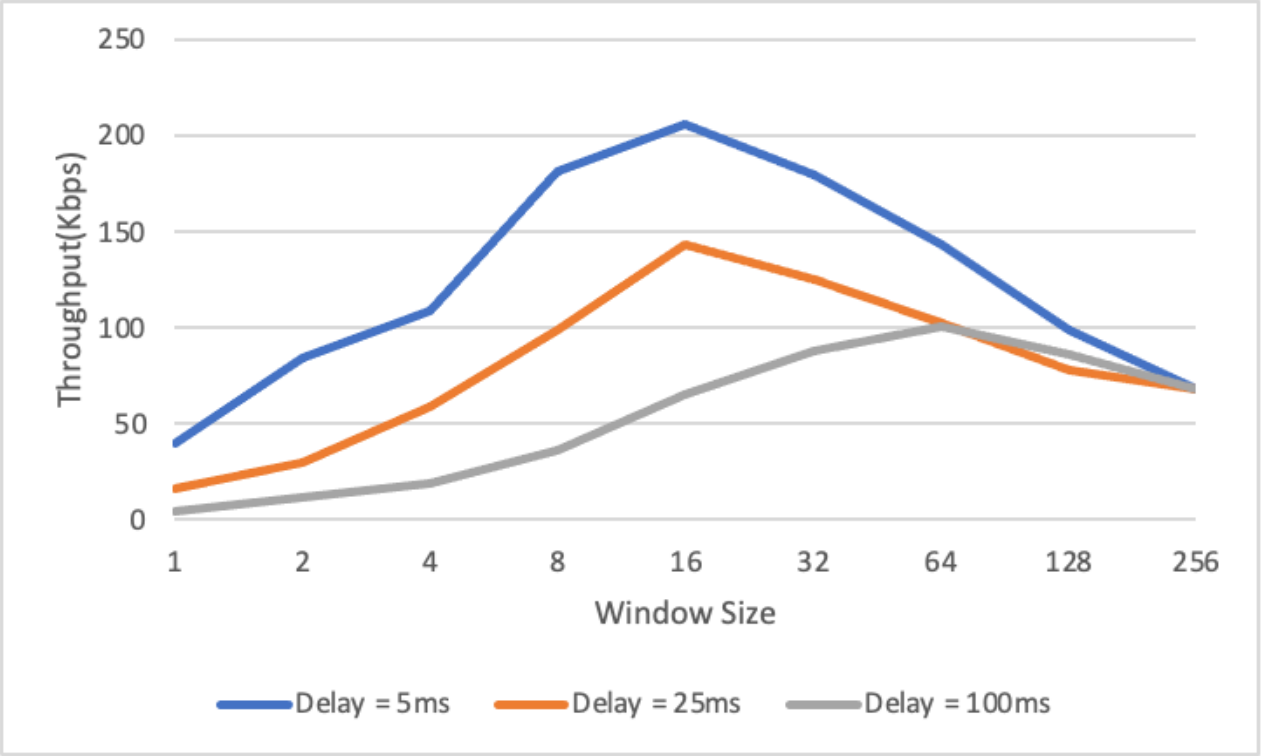
The average number of re-transmissions tends to be stable around 100 times after 15ms retransmission timeout. Because the re-transmissions depend on the packet loss rates, it will not influence the re-transmission numbers after 15ms timeout. The average throughput decreases gradually as the retransmission timeout increase. This is because in a high timeout condition, the packet must wait for a timeout then sending the lost packet.

The optimal timeout value would be 15ms or 20ms, and the best throughput are around 75KB/s.

Question 3 – Experimentation with Go-Back-N. For each value of window size, run the experiments for **5 times** and write down **average throughput**.

	Average throughput (Kilobytes per second)		
Window Size	**Delay = 5ms**	**Delay = 25ms**	**Delay = 100ms**
1	39.8	16.7	4.8
2	84.1	30.1	12.2
4	108.8	58.9	18.8
8	181.4	98.7	36.7
16	205.8	143.1	65.4
32	180.1	125.2	88.2
64	143.8	102.2	100.8
128	98.9	78.2	85.9
256	68.1	68.2	68.4

Create a graph as shown below using the results from the above table:



Question 4 – Discuss your results from Question 3.

For the 5ms time delay, I choose 20ms because it's the optimal value from previous question. For 25ms time delay, I chose 55ms, 25*2+5ms. because it takes at least 50ms of RTT and some processing time for a packet to arrive and get an ACK. I choose 205ms because of same reason. 200ms RTT+5ms processing time. The best performance is using 5ms time delay and window size is 16. And then 25ms delay is better than 100ms. Because in low time delay condition, packets can get and send packets faster, thus total time spending is less, the throughput is higher.

When the window size is less than 16, the throughput increases as the window size increases. Because at a time delay of 5ms, most of the window is used and the time delay is reasonable. Therefore, a window size of 16 and a time delay of 5ms is the best. As the window size increases (greater than 16) and the time delay doesn't change, most of the window is useless because it will store outdated packets and ACKs, which does not help to improve the throughput.

In addition, as the time delay increases from 5ms to 25ms or 100ms. the gradient of throughput decreases, which means that throughput does not decrease significantly with increasing time delay. This is because when the time delay is higher, the socket has more time to process the ACKs in the window. Therefore, the data in the window will be more useful than a short time delay of same window size.

Therefore, the selection of the optimal window size should also take into account the time delay factor.

Question 5 – Experimentation with Selective Repeat. For each value of window size, run the experiments for **5 times** and write down **average throughput**.

	Average throughput (Kilobytes per second)
Window Size	**Delay = 25ms**
1	15.3
2	29.6
4	55.3
8	88.7
16	155.1
32	164.2

Question 6 - Compare the throughput obtained when using "Selective Repeat" with the corresponding results you got from the "Go Back N" experiment and explain the reasons behind any differences.

In this question, I using the same retransmission timeout 50ms. When the window size smaller than 8, GBN and SR have similar performance. As the window size greater than 8, SR has better performance than GBN. When the window size is small, the difference between selecting packets to send or sending all packets directly can be ignored. When the window size becomes larger, sending all packets will waste time because most of the packets have already received ACKs. but SR can avoid this situation by selecting those packets that time out so that it can avoid sending successfully received packets and save time. In general, SR will have better performance when the window size increases.

Question 7 – Experimentation with *iperf*. For each value of window size, run the experiments for **5 times** and write down **average throughput**.

Window Size (KB)	Average throughput (Kilobytes per second) Delay = 25ms
1	7.5
2	18.9
4	26.1
8	43.2
16	55.4
32	68.9

Question 8 - Compare the throughput obtained when using "Selective Repeat" and "Go Back N" with the corresponding results you got from the *iperf* experiment and explain the reasons behind any differences.

In general, Go-Back-N and Selective Repeat have better performance than using iperf when using higher window sizes. Because iperf requires a TCP connection, and TCP requires a three-way handshake for security. It takes longer than UDP.
In addition, TCP has a congestion control. When packet loss is high, some packets are congested in the channel. the TCP connection must drop some packets to ensure that the channel is not congested. This will prevent iperf from achieving high performance at high packet loss rates.