CN-3530/CS 301 Assignment 2

1. Stop and Wait Protocol

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 the average **number of retransmissions** and **average throughput**.

Retransmission timeout (ms)	Average number of re-transmissions	Average throughput (Kilobytes per second)
5	1212	78.95
10	505	74.78
15	134	66.73
20	139	65.47
25	120	53.44
30	133	53.22
40	112	58.90
50	97	55.44
75	102	47.36
100	104	44.37

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

Impact of Retransmission Timeout on Retransmissions:

1.A shorter retransmission timeout (e.g., 5 ms) results in a significantly higher number of retransmissions. For example, at 5 ms, there are 1212 retransmissions.

2.As the timeout value increases, the number of retransmissions decreases drastically. For instance, at 10 ms, retransmissions drop to 505, and by 50 ms, they decrease further to 97.

3.This trend suggests that a short timeout leads to unnecessary retransmissions, as packets might still be on the way, but the protocol times out too quickly and resends them.

Impact of Retransmission Timeout on Throughput:
1.Throughput is highest at the shortest timeout of 5 ms, with a throughput of 78.95 KB/s, but this also has the highest retransmissions.
2.As the timeout increases, throughput gradually decreases. For example, at 25 ms, throughput drops to 53.44 KB/s, and at 100 ms, it further reduces to 44.37 KB/s.
Optimal Timeout Value:
1.From a communication efficiency perspective, the optimal timeout should balance minimizing retransmissions while maintaining reasonable throughput.
2.Timeout values in the range of 40-50 ms seem optimal. For example:
At 40 ms, retransmissions are at 112, and throughput is 58.90 KB/s.
At 50 ms, retransmissions reduce to 97, and throughput is 55.44 KB/s.
Conclusion:
The optimal retransmission timeout value for efficient communication is around **40-50 ms**. This range minimizes unnecessary retransmissions while maintaining a reasonable throughput, making it ideal for balancing reliability and performance.

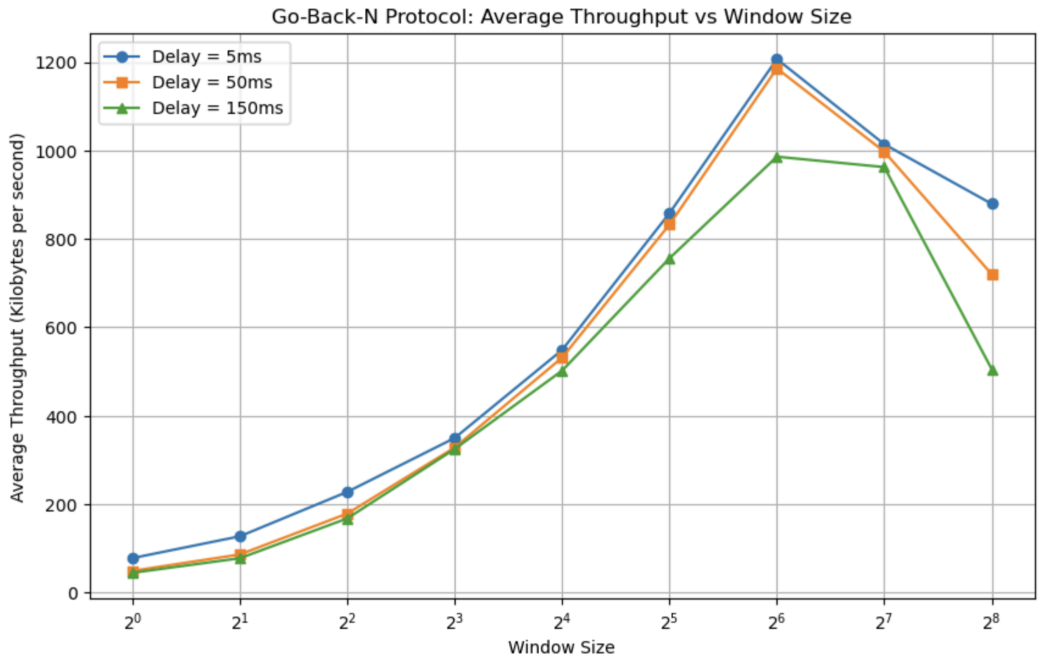
2. Go back N Protocol

Question 1 – Experimentation with Go-Back-N. For each value of window size, run the experiments **5 times** and write down the **average throughput**.

	Average throughput (Kilobytes per second)		
Window Size	**Delay = 5ms**	**Delay = 50ms**	**Delay = 150ms**
1	77.37	48.55	45.29
2	127.88	85.72	78.25
4	228.43	179.16	168.27
8	350.09	328.40	326.11
16	549.02	531.75	502.24
32	855.27	832.54	756.26

64	1206.59	1185.43	985.91
128	1015.18	997.24	962.37
256	878.39	720.19	505.86

Create a graph similar to the one shown below using the results from the above table: (Edit: change delays to 5ms, 50ms and 150 ms as mentioned in the assignment statement)



Question 2 – Discuss your results from Question 1.

Throughput Increase with Window Size:

1.As the window size increases, the average throughput also increases across all delays. For instance, at a delay of 5ms, throughput rises from 77.37 KB/s (window size 1) to a peak of 1206.59 KB/s (window size 64).

2.This is expected, as a larger window size allows more packets to be sent before requiring an acknowledgment, leading to better utilization of the network.

Impact of Delay on Throughput:

1The throughput is highest when the delay is shortest (5ms), and it decreases as the delay increases. For example, with a window size of 4, the throughput is 228.43 KB/s at 5ms delay, 179.16 KB/s at 50ms delay, and 168.27 KB/s at 150ms delay.

2.This behavior is due to the Go-Back-N protocol's reliance on acknowledgments. Higher delays mean longer wait times for acknowledgment, reducing the effective throughput.

Diminishing Returns with Larger Window Sizes:
1.While increasing the window size improves throughput, there is a point of diminishing returns. For instance, when the window size increases from 128 to 256, the throughput decreases across all delays.
2.This may occur because larger window sizes increase the risk of packet loss, requiring retransmissions that hurt overall throughput. Additionally, network conditions and protocol limitations can cause efficiency to drop as the window size continues to grow.
Optimal Window Size:
1.From the data, it seems that a window size of 64 offers the best balance for maximizing throughput, especially at lower delays. Beyond this point, the throughput starts to decrease, indicating an optimal window size for this specific network configuration.
Conclusion: The data demonstrates that in the Go-Back-N protocol, both the window size and delay significantly affect throughput. A larger window size generally improves throughput up to an optimal point, but excessive window sizes may reduce efficiency due to increased retransmissions. Lower delays yield better throughput, showing the importance of minimizing network latency in achieving optimal performance in Go-Back-N-based systems.