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

# Assignment 2 Results Sheet

| | |
|---|---|
| **Forename and Surname:** | Arbnor Bregu |
| **Matriculation Number:** | s1832263 |

**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 | 951.8 | 68 |
| 10 | 619.8 | 63 |
| 15 | 163.4 | 60 |
| 20 | 108.8 | 57 |
| 25 | 93.4 | 56.2 |
| 30 | 98.2 | 54 |
| 40 | 96 | 51.8 |
| 50 | 100.2 | 48.2 |
| 75 | 92.8 | 43.6 |
| 100 | 97 | 38.5 |

**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 drastically decreases the number of retransmitted packets when the timeout goes from 5 to 20, where it drops by an average of 840. However from a timeout of 25 to 100 the number of retransmitted packets seems to fluctuate between the different timeouts, for example a timeout of 75ms has less retransmissions than a timeout of 100ms, this indicates that past a timeout of 25ms the number of retransmitted packets does not go though any more major changes as it tends to stay within a range of 90 - 100 retransmissions.

Increasing the timeout has a negative correlation with the average throughput, this is because by increasing the time the socket stays active waiting for the correct ACK packet to be received, this increases

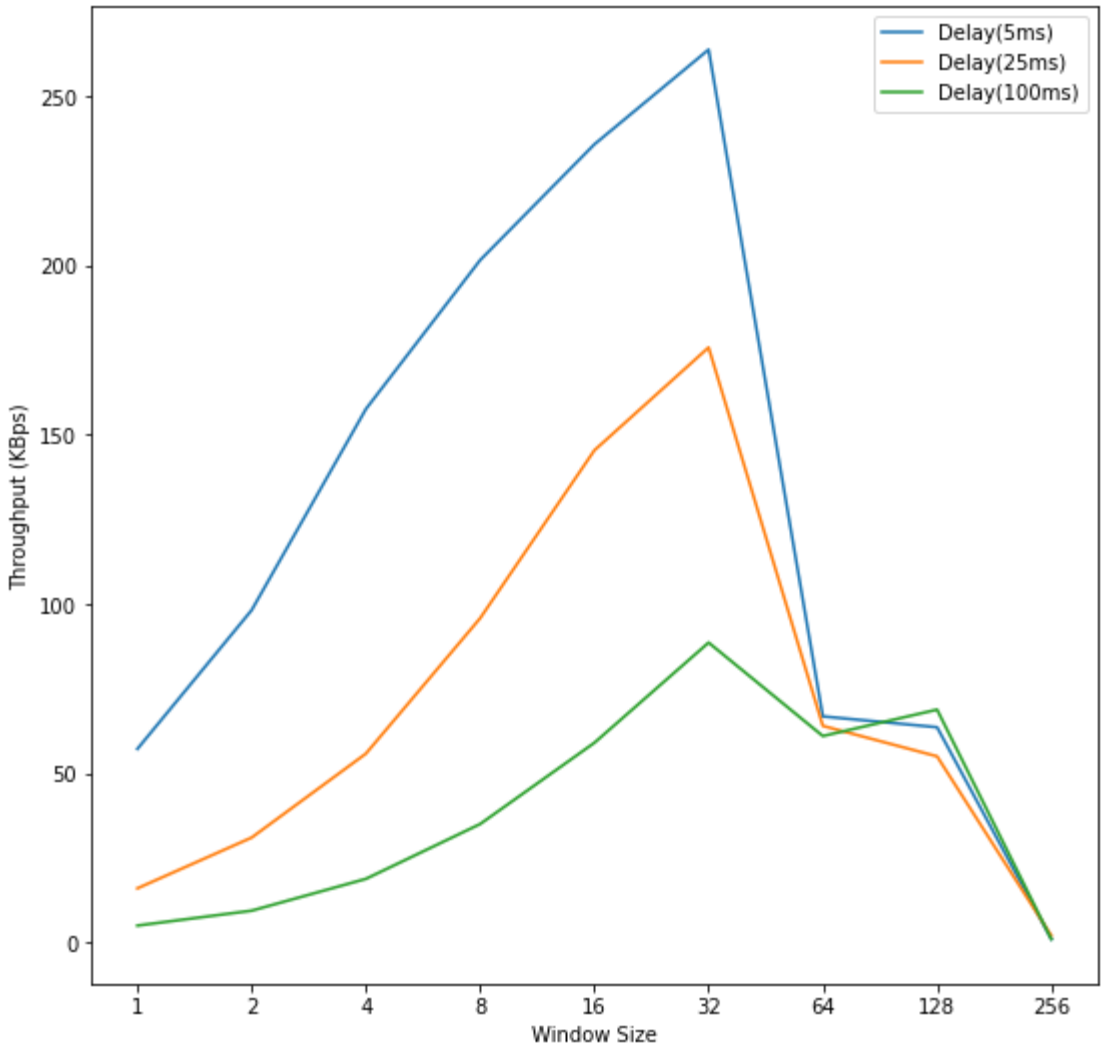
the overall time it takes for the packet to be delivered to the receiver therefore decreasing the throughput because the size of the file is a constant.

I believe the optimal timeout value should be 25ms as it has a sufficiently high throughput whilst also keeping a lower average number of retransmitted packets.

**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 | 57.2 | 16 | 5 |
| 2 | 98.2 | 31 | 9.4 |
| 4 | 157.6 | 55.8 | 18.8 |
| 8 | 201.6 | 95.8 | 35 |
| 16 | 235.8 | 145.4 | 59 |
| 32 | 263.8 | 175.8 | 88.6 |
| 64 | 66.8 | 64 | 61 |
| 128 | 63.6 | 55 | 68.8 |
| 256 | 1 | 2 | 1 |

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




**Question 4** – Discuss your results from Question 3.

From this graph it is evident that increasing the window size past 32 has a negative impact on the throughput, this is most likely due to if a packet in the window has not been transmitted correctly or it has not been ACKed, every packet after and including, that specific packet must be retransmitted. This leads to a lot of packets being retransmitted which will cause further delays because many packets will be resent therefore reducing the throughput.

An increased propagation delay time also drastically reduces the throughput, this is because it takes packets longer to travel through the network and be received at the receiver end, due to the increased in delay the overall time it takes all packets to be transmitted takes longer therefore heavily reducing throughput.

If window size > 32 and a larger delay (>5ms), the throughput suffers the most.

**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 |
| 2 | 30.4 |
| 4 | 62.6 |
| 8 | 123.6 |
| 16 | 219.6 |
| 32 | 361.6 |

**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.

For my results with a window size of 1 or 2 the throughput is almost the same, however with a window size of 4 or above Selective Repeat is a much more efficient method as it has a higher average throughput value. This is most likely due to SR only resending the packets in the window that have timed out, whilst all the other packets that have been correctly ACKed do not need to be resent.

However with Go-Back-N if a packet has timed-out (an ACK has not been received for it) then all the consecutive packets following that packet in the window will also have to be sent again, even if their ACK had been received by the socket, this is inefficient when compared to Selective Repeat.

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

| | Average throughput (Kilobytes per second) |
|---|---|
| **Window Size (KB)** | **Delay = 25ms** |
| 1 | 12 |
| 2 | 12.4 |
| 4 | 12.5 |
| 8 | 24.7 |
| 16 | 94.8 |
| 32 | 100.7 |

**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.

iperf has a much slower throughput than both selective repeat and Go-Back-N, at all window sizes, this is most likely due to iperf having to establish a connection with the server and client as it guarantees delivery of every packet. Therefore each packet must be transmitted and correctly received before the next packet can be sent. This will most likely lead to longer run times for the same file size, and therefore a reduced throughput.