Computer Communications and Networks (COMN) 2016/17, Semester 2

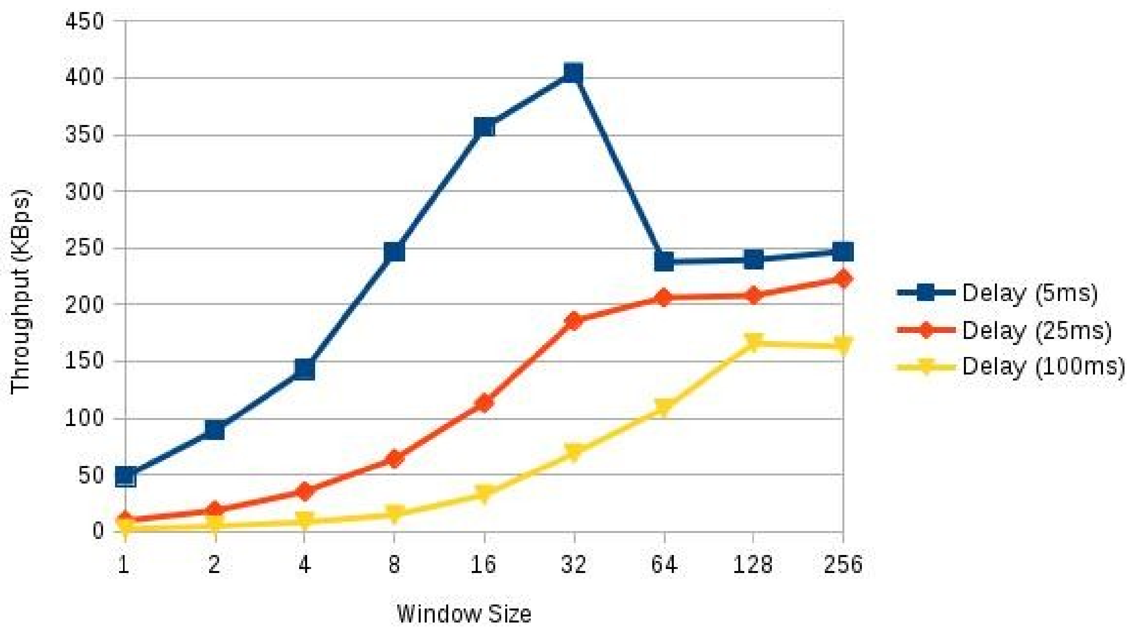
Assignment Part 2 Results Sheet

Forename and Surname:	Kendeas Theofanous
Matriculation Number:	s1317642

Question 1 – Experimentation with Go-Back-N:

	Throughput (Kilobytes per second)		
Window Size	**Delay = 5ms**	**Delay = 25ms**	**Delay = 100ms**
1	48.76	9.9	2.42
2	89.12	18.55	4.68
4	142.57	35.25	8.48
8	246.63	64.14	14.77
16	356.66	113.36	32.36
32	404.58	185.78	68.96
64	237.69	206.38	108.16
128	239.44	208.33	165.99
256	246.87	223.07	163.19

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



Question 2 – Discuss your results from Question 1.

The values of the table are the average values from a sample of five tries on every window size for every delay value. In order to take these results, I used the optimal retransmission timeout from the last part of the coursework, which was 25ms. Looking at the results and the graph, it can be deducted that the throughput has the lowest value at window size 1 for every delay value and is almost doubled after every window size change until it reaches the peak value. As can be seen from the results, the delay greatly affects the throughput, the highest throughput being at delay 5ms and window size 32, reaching 404 KB/s, and the lowest being at delay 100ms and window size 1 at 2.4 KB/s.

Looking at the graph, the line of delay 5ms is the steepest from window size 1 to 32, starting from a value of 48 KB/s and as window size increases, throughput tends to almost double, reaching a peak value of 404 KB/s. At window size 64, throughput falls to 238 KB/s and after that it’s roughly the same for window sizes 128 and 256.

For delay of 25ms, throughput is the lowest at window size 1, being 9.9 KB/s and roughly doubles as window size increases reaching 32 and value 186 KB/s, and for the rest of the window sizes it is flactuating around 210 KB/s.

For delay of 100ms, throughput is the lowest at window size 1, being 2.4 KB/s and roughly doubles as window size increases reaching 128 and value 166 KB/s. For window size 256, throughput remains roughly the same.

In conclusion, greater window sizes are faster using Go Back N. Packet loss being at 0.5% is not frequent, therefore greater window size allows more packets to be sent and with the optimal retransmission time it is faster.

Question 3 – Experimentation with Selective Repeat

	Throughput (Kilobytes per second)
Window Size	**Delay = 25ms**
1	7.89
2	15.1
4	29.62
8	56.32
16	112.97
32	183.53

Question 4 - 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.

The same procedure was used to take the results of this table in order to have a fair comparison of the two principles, i.e. the values of the table are the average values from a sample of five tries on every window size, using the same retransmission timeout of 25ms. Comparing the two principles, it can be seen that Selective Repeat is slightly slower for small window sizes. I would assume that this would stop being the case even for window size 32, but my results show thate even for that window size Selective Repeat is slightly slower. This happens because the window size is not big enough to make the difference. As Selective Repeat holds a timer for every packet and retransmits only that packet as soon as its timer expires, given that an acknowledgement was not received, makes a greater difference on throughput for window sizes 64, 128 and 256. Go Back N holds only one timer at the start of the window and waits for all acknowledgements and if there was a packet loss in the window, the window after the last acknowledgement was received needs to be retransmitted. In order to ensure my implementation was correct, I tested Selective Repeat for these window sizes and the difference of throughput from Go Back N is slightly greater for window size 64, at around 212 KB/s and far greater for window size 128 and 256 at around 255 KB/s and 300 KB/s respectively.

Question 5 – Experimentation with *iperf*

Window Size (KB)	Throughput (Kilobytes per second) Delay = 25ms
1	14.0
2	17.1
4	39.3
8	65.4
16	75.5
32	93.2

Question 6 - 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.

The iperf experiment opens a TCP connection between the receiver and the sender. A TCP connection is considered a hybrid of Go Back N and Selective Repeat protocols. The main advantage of TCP is its flow-control service. The hosts of the connection have a receive buffer where the correct and in sequence bytes are placed at. If the application is slow at reading data, the sender side might overflow the receiver's buffer. TCP handles that by matching the speed of sender sending the data and the receiver's speed of reading the data. For a fair comparison, the values of the table are the average values from a sample of five tries on every window size. Looking at the results, throughput using iperf is slightly greater for windows 1 to 8 compared to either of the two principles. For the rest of window sizes, iperf is noticeably slower, nearly half for window size 16 and even less than half for window size 32, at around 75 Kb/s and 93 KB/s respectively. The reason for this is the buffer mentioned earlier. By default, the buffer size for either of the TCP connection will be 8 KB therefore restricting the amount of data the sender will send to the receiver, making the throughput smaller for window sizes greater than 8 KB.