Computer Communications and Networks

Forename and Surname:	Nicholas Swafford
Matriculation Number:	s0827481

Question 1

The following is a tabulation of experimental results detailing the impact of retransmission timeout on the number of retransmissions with the stop-and-wait protocol.

Retransmission Timeout (ms)	Number of Retransmissions	Throughput (KBps)
10	16542	23.42
20	10671	23.15
30	4212	21.37
40	1337	20.29
50	622	19.31
60	637	18.33
70	644	17.61
80	616	17.08
90	605	16.60
100	601	15.16

Question 2

How does the throughput behave when the retransmission timeout increases and why? What is the optimal value for retransmission timeout?

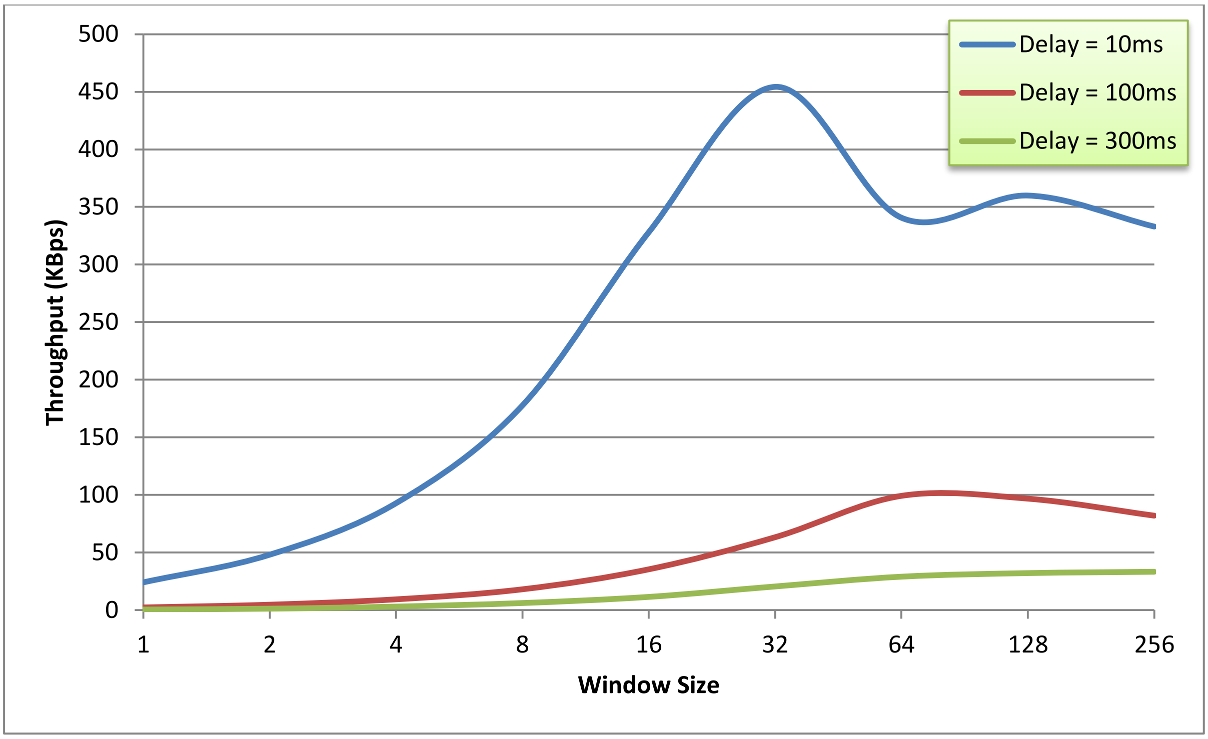
The Stop-And-Wait protocol prohibits the transfer of new packets until the current one has been acknowledged. As a consequence no data is processed or transferred in the time it takes for the stalled packet to timeout thereby decreasing the overall throughput of the protocol. As the timeout increases the number of retransmissions decreases since unnecessary retransmissions, that were due simply to processing or network latency rather than packet loss, are eliminated.

The optimal value for retransmission timeout will be one that balances speed with amount of data transferred. Throughput steadily decreases but the number of retransmissions starts to plateau near the 50 millisecond mark so I have chosen it as the optimal value.

Question 3

The following table of results, and subsequent graphical representation, were built from experiments carried out with the Go-Back-N protocol.

	Throughput (Kilobytes per second)		
Window Size	Delay = 10ms	Delay = 100ms	Delay = 300ms
1	24.17	2.41	0.62
2	48.17	4.80	1.34
4	92.89	9.45	3.16
8	177.70	18.15	6.23
16	328.04	35.50	11.56
32	454.24	63.34	20.62
64	340.64	99.26	29.11
128	359.91	96.84	32.20
256	332.98	82.07	33.37



Question 4

Explain the results from Question 3.

Due to the increase in network transmission delay the timeout had to be scaled appropriately. Using the optimal value found in Q1 (50ms) for the configuration with a one way delay per pipe of 10ms the experiment with a delay of 100ms had a timeout of 500ms and the one with a 300ms delay had a timeout of 1500ms.

The throughput decreased significantly as the timeout increased due to the combination of network delay and, as expected, the retransmission timeout. The window size also played a significant role in the throughput's variation. Quite noticeable in the graph of the results with a 10ms delay is a throughput peak at a window size of 32, after which it takes a short dive. The Go-Back-N protocol is forced to retransmit all packets in the window once the oldest packet in the window times out. With a larger window size, not only is there a higher chance of packet loss forcing a retransmission of large amounts of data, but the receiver may be flooded with the arrival of large amounts of data at once, and fail to successfully process and transmit acknowledgements for certain packets or take longer than the given timeout to process it all (although the latter grows more unlikely as timeout increases).

Question 5

The following table of results was built from experiments carried out with the Selective Repeat protocol.

	Delay = 100ms	
Window Size	Throughput (KBps)	Number of Retransmissions
8	18.79	108
16	36.66	162
32	70.17	256
64	122.38	598
128	207.46	1054
256	308.07	2051

Question 6

Compare the throughput obtained when using Selective Repeat with the corresponding results achieved from the Go-Back-N experiments and explain the reasons behind any differences.

One of the advantages of Selective Repeat is that it can buffer any assortment of data within the window and only have to retransmit the packets that were missing an acknowledgement after the timeout. This eliminates the need for the retransmission of a large number of packets. As we can see in the tables, the throughput for both protocols at smaller window sizes is very similar. The advantages of Selective Repeat become evident in the results as the window size increases.

Question 7

The following table of results was built from experiments carried out with iperf.

	Delay = 100ms	
Window Size	Throughput (Kbits per second)	Throughput (Kbytes per second)
8	150.0	18.75
16	226.8	20.25
32	288.0	36.00
64	301.8	37.73
128	426.2	53.28
256	386.2	48.28

Question 8

Compare the throughput obtained from the previous two protocols with the corresponding results achieved from the iperf experiment and explain the reasons behind any differences.

iperf measures the throughput, in kilobits, of an implementation of the TCP protocol on a portion of provided data. TCP transmits and receives data as a stream, handles error checking and handshakes, and has a larger header in comparison to UDP. This means that less data is transferred per packet at slower rates (due to all the processing being handled at the network level), and the loss of a packet during the transfer can halt the stream. In comparison, UDP accepts all incoming data and handles all the processing at the application layer. The results, once converted into kilobytes per second, over a delayed and lossy channel demonstrate that both of the refined UDP protocols significantly outperform TCP as the window size increases. The Go-Back-N protocol approaches the TCP protocol as window sizes get larger since it begins behaving similarly. Since the start of the window moves after the earliest packet in the window is acknowledged, once the window size is the size of the file being transferred it behaves like an inefficient stream.