Modern Networking Concepts

PA 2 Report

Reliable Transport Protocols

I have read and understood the course academic integrity policy.

Introduction

This report gives the brief implementation logic and performance graphs for reliable transfer protocols Alternating Bit Protocol, Go-back-N and Selective repeat.

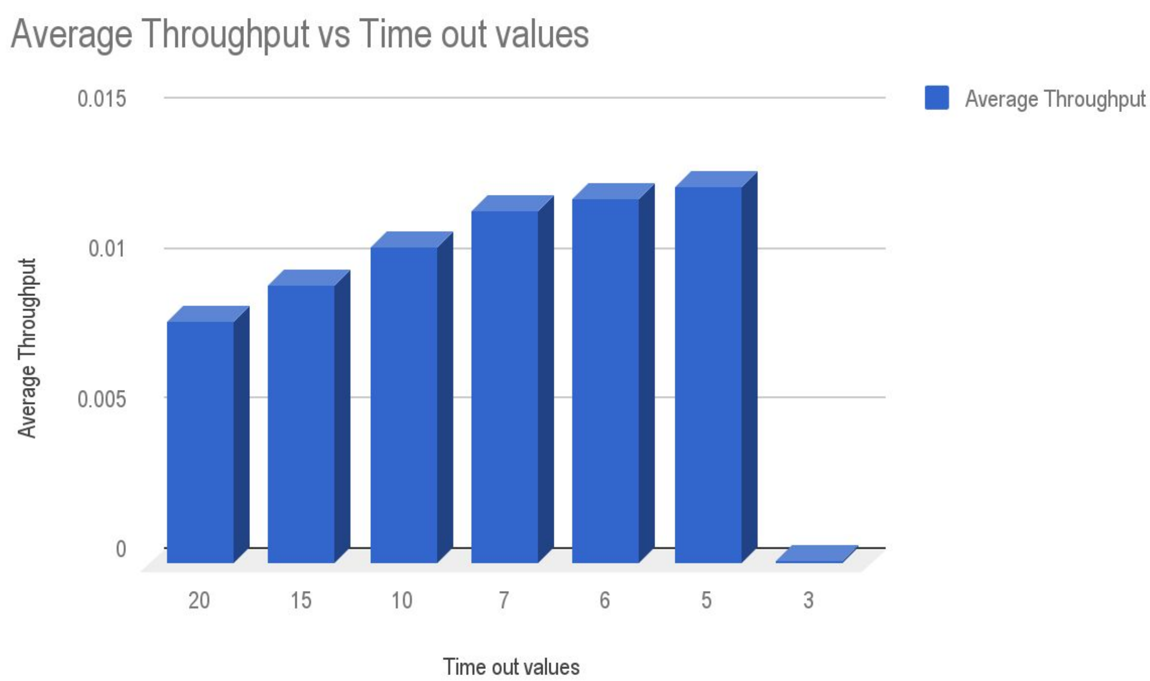
Time out scheme

For choosing the best time out value, I ran each of my protocols for different time out values and noted the average throughput for loss 0.1, 0.2, 0.4, 0.6 and 0.8. Whichever time out gives us the best throughput, it has been chosen as the value.

Time Out Scheme for ABT

Time out value	Average throughput for loss (0.1, 0.2, 0.4, 0.6, 0.8)
20.0	0.008054
15.0	0.009248
10.0	0.010564
7.0	0.011762
6.0	0.01212
5.0	0.012578
3.0	0.0000916

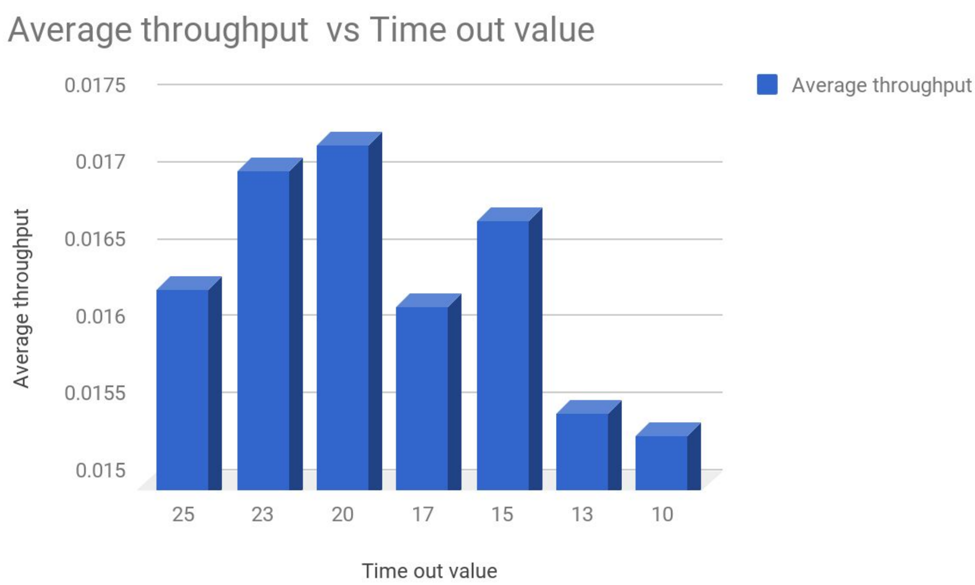
As it can be seen that choosing a fixed timeout value 5.0 gives the best performance for ABT. This is expected because when I calculated the total roundtrip time of the packet going to the receiver and the time for the acknowledgment to come back. It was around 5.0.



Time Out Scheme for GBN Window 10

Time out value	Average throughput for loss (0.1, 0.2, 0.4, 0.6, 0.8)
25.0	0.01625
23.0	0.01703
20.0	0.01719
17.0	0.01614
15.0	0.01670
13.0	0.1545

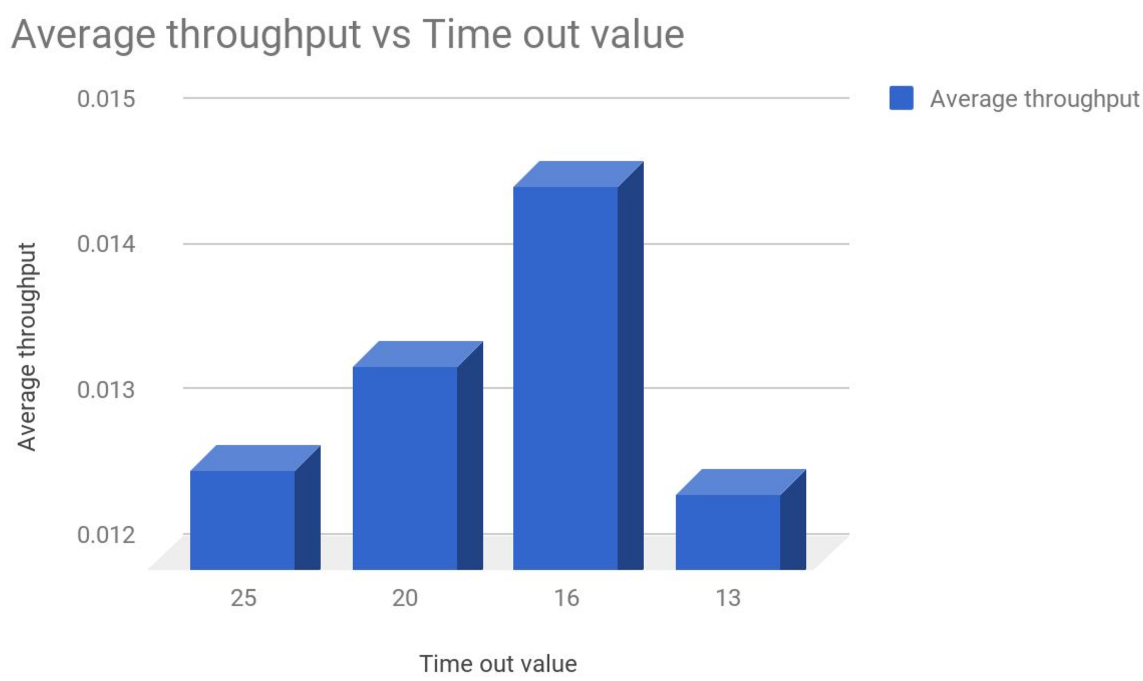
10.0	0.01531



For GBN of window size 10 I observed that my protocol performs better with time out value set to 20.0.

Time Out Scheme for GBN Window 50

Time out value	**Average throughput for loss (0.1, 0.2, 0.4, 0.6, 0.8)**
25.0	0.01261
20.0	0.01333
16.0	0.01456
13.0	0.01244

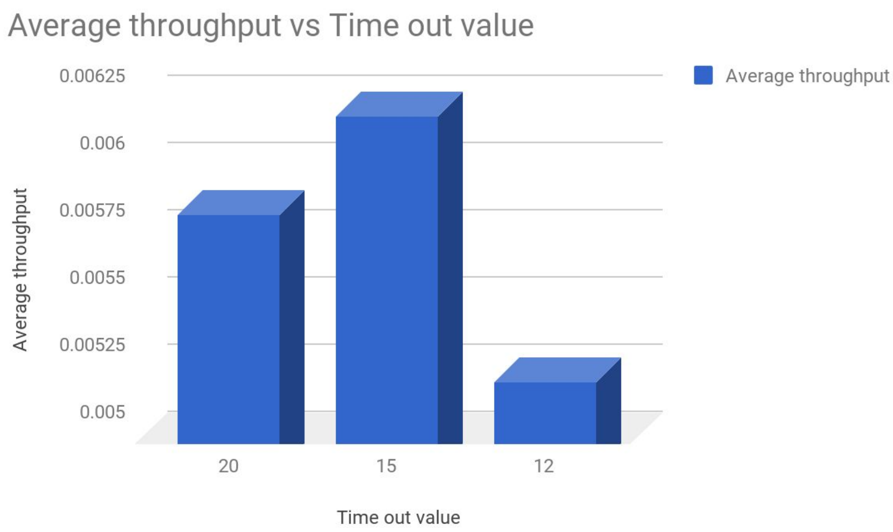


For GBN of window size 50, I observed that my protocol performs better with time out value set to 16.0.

Time Out Scheme for SR Window 10

Time out value	**Average throughput for loss (0.1, 0.2, 0.4, 0.6, 0.8)**
20.0	0.005822
15.0	0.00619
12.0	0.005200

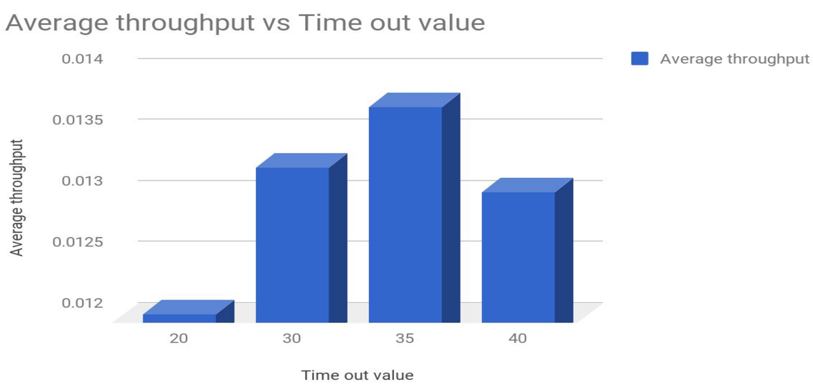
For selective repeat, I am not using a adaptive time-out and it remains fixed at some static value. When I checked for different values, I found out that performance was best at timeout 15.0 for window size 10.



Time Out Scheme for SR Window 50

Time out value	Average throughput for loss (0.1, 0.2, 0.4, 0.6, 0.8)
20.0	0.01202
30.0	0.01322
35.0	0.01372
40.0	0.01302

Selective repeat with window 50 performed best at timeout value 35.0.



Multiple Timer scheme for Selective repeat

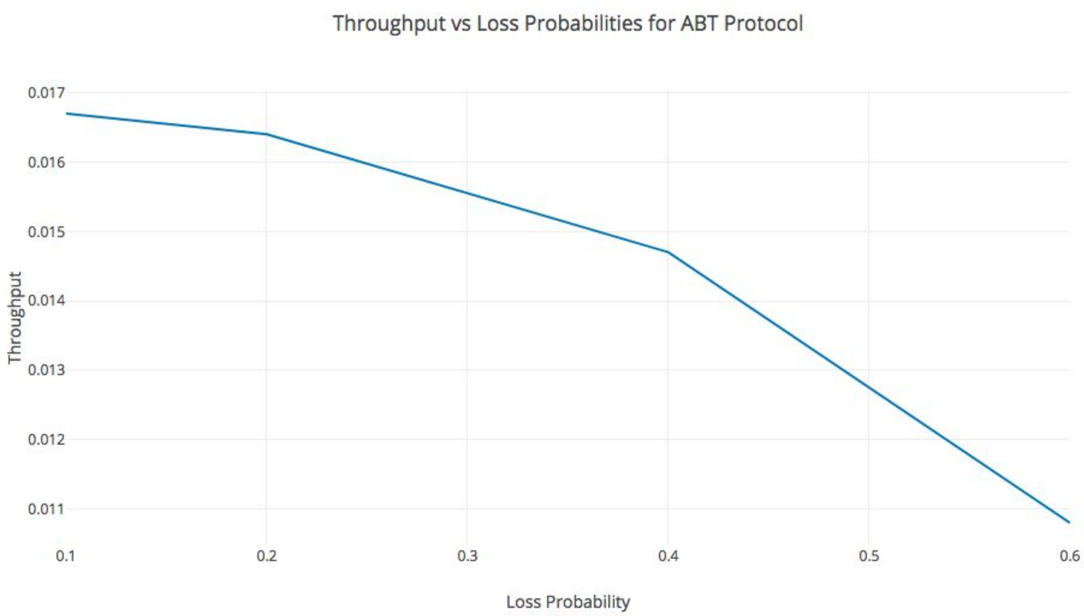
For selective repeat, since we have a single timer I maintained a vector list for each packet. The index of this vector list corresponds to the seq no of each packet. Whenever I send a packet from sender side, I note the current time using get_sim_time() inside my timer. The set the timer at some static value "TimeOutVal" If this packet comes back in time, I mark it as acknowledged and stop the timer. I then start the timer again with the oldest unacknowledged packet. This time I start the timer with value "TimeOutVal - (currentTime - timer[i])" time because, this much time has been elapsed since this packet was last sent. I then update the timer[i] with current time.

Everytime I start a timer, I note the packet with started this timer. If it times out, I simply resent that particular packet which started it. After that I start the timer again with oldest unack packet.

In the A_input() I also check my timer list for packets which have currentTime() > timer[i] + RTT. If that's the case, I resend that packet.

ABT Protocol

Loss Probability	ABT
0.1	0.0167
0.2	0.0164
0.4	0.0147
0.6	0.0108



For ABT, I observed that throughput falls as the loss probability increases. This is expected because ABT is a very simple protocol. In a lossy channel, The stop and wait step in that case makes the protocol wait for very long during a loss. Hence, a very less utilization of channel.

GBN Protocol

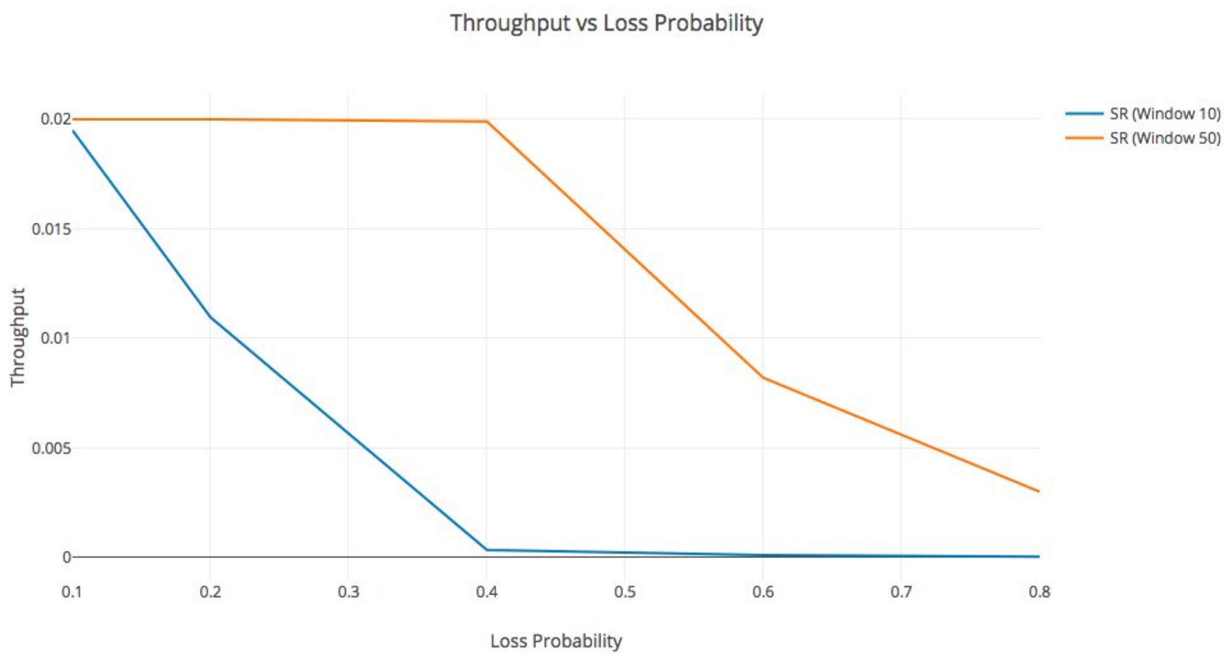
Loss Probability	GBN (Window 50)	GBN (Window 10)
0.1	0.02014	0.0199
0.2	0.02025	0.02012
0.4	0.0199	0.01766
0.6	0.0098	0.0193
0.8	0.0027	0.00897

I observed that when loss were low GBN with window 50 performed slightly better than window 10. However, when the losses were high GBN with window 50 performed worse.

Thus low window sizes are better in a lossy environments for GBN. I also observed that in a highly corrupted environment the GBN protocol takes a very long time. Because the window does not move forward and there a lot of retransmissions. A better GBN implementation changes timeout values for different window sizes.

SR Protocol

Loss Probability	SR (Window 10)	SR (Window 50)
0.1	0.0195	0.0200
0.2	0.01096	0.0200
0.4	0.00034	0.0199
0.6	0.00011	0.0082
0.8	0.00004	0.003

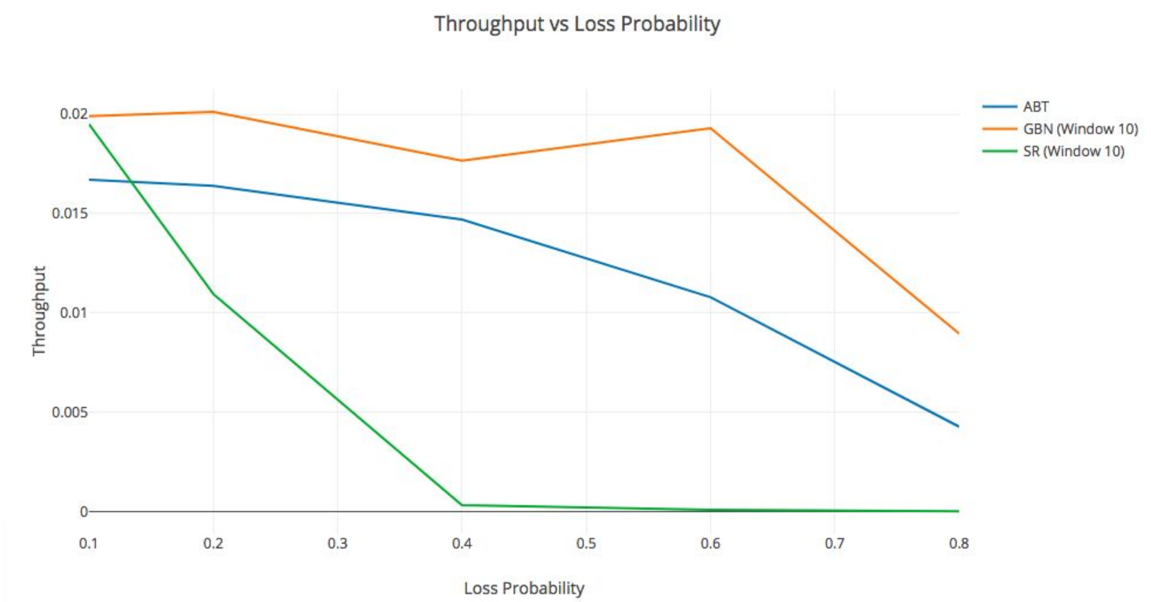


SR protocol was worse for lower window sizes. However on increasing the size to 50, it performed on par with GBN. For selective repeat, The window size w_r need only be larger than the number of *consecutive* lost packets that can be tolerated (Reference - https://en.wikipedia.org/wiki/Sliding_window_protocol#Selective_repeat). For loss probability greater than 0.4, there may be a lot of consecutive packets being lost which results in poor performance. Performance of SR depends on window sizes. Higher window sizes are better in this case.

Experiment 1 -

1) **Throughput (ABT, GBN (W 10), SR (W 10)) vs Loss Probability 0.1, 0.2, 0.4, 0.6, 0.8.**

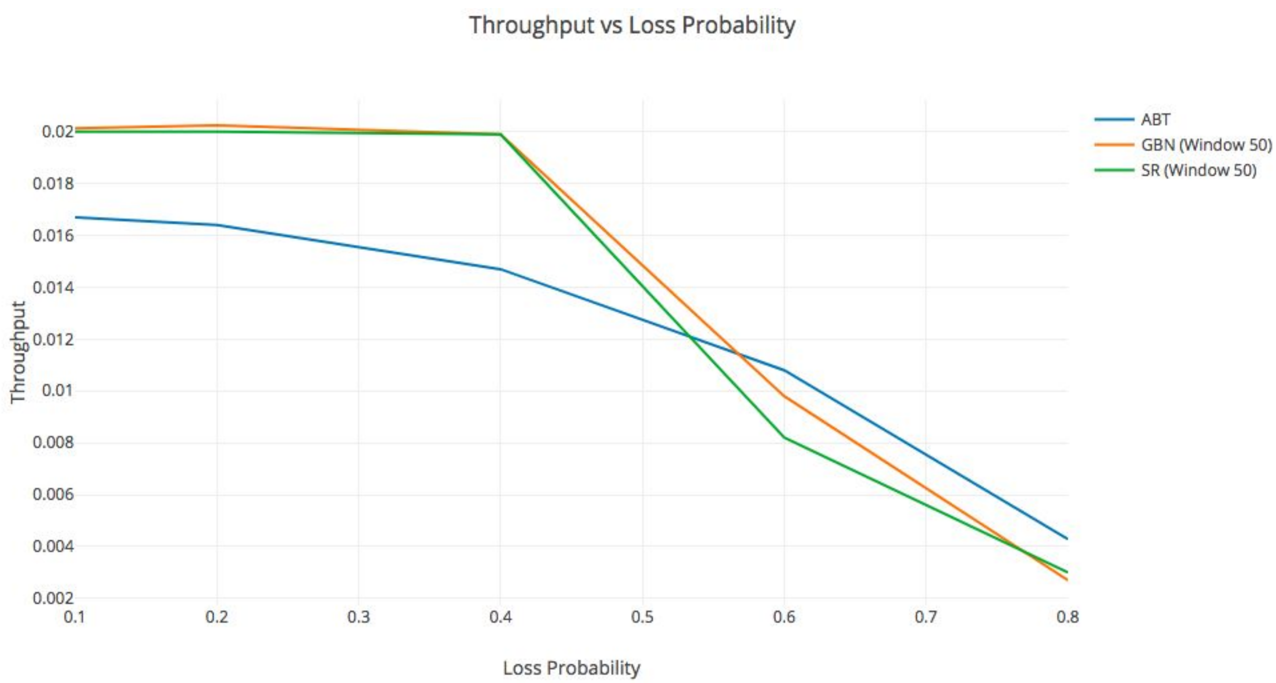
Loss Probability	**ABT**	**GBN (Window 10)**	**SR (Window 10)**
0.1	0.0167	0.0199	0.0195
0.2	0.0164	0.02012	0.01096
0.4	0.0147	0.01766	0.00034
0.6	0.0108	0.0193	0.00011
0.8	0.00429	0.00897	0.00004



As it can be seen, GBN performs the best for window size 10. This is expected as GBN is better for less window size. Selective repeat suffers from consecutive packet losses greater than the window size. Hence its performance is even worse than ABT.

2) Throughput (ABT, GBN (W 50), SR (W 50)) vs Loss Probability 0.1, 0.2, 0.4, 0.6, 0.8.

Loss Probability	ABT	GBN (Window 50)	SR (Window 50)
0.1	0.0167	0.02014	0.0200
0.2	0.0164	0.02025	0.0200
0.4	0.0147	0.0199	0.0199
0.6	0.0108	0.0098	0.0082
0.8	0.00429	0.0027	0.003

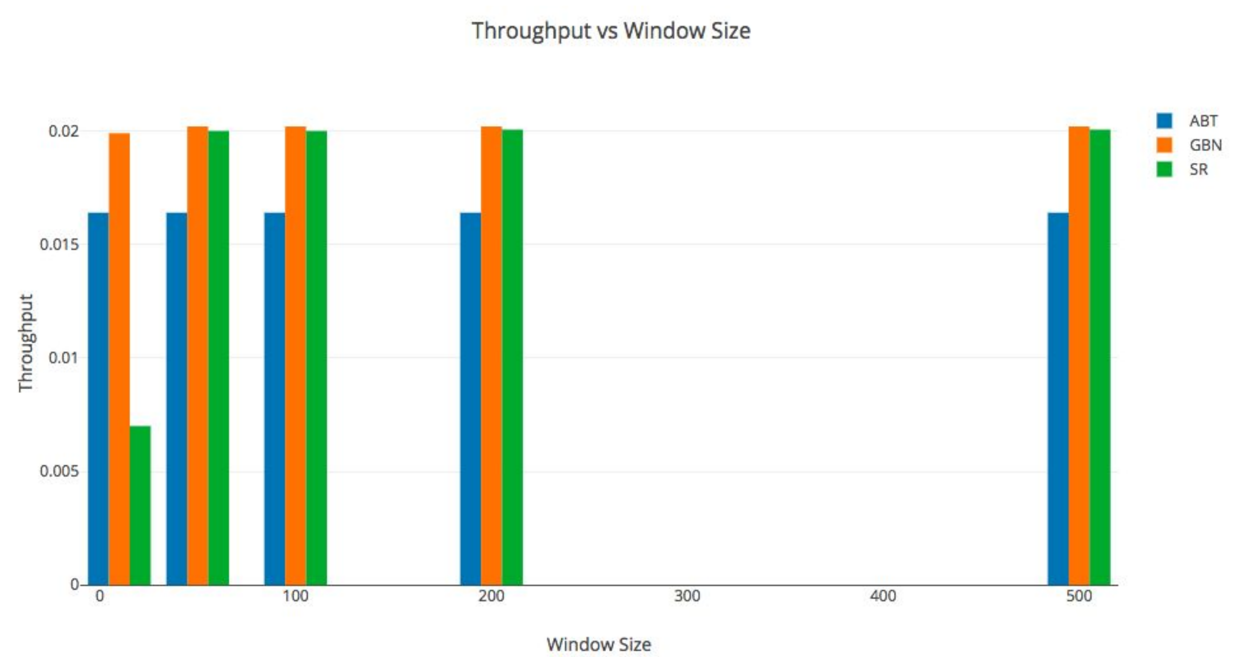


For small loss probability, GBN and SR protocols were better than ABT. However, for larger losses ABT performed slightly better than GBN and SR. In terms of running time and efficiency, GBN was very slow for larger losses and corruption. If a packet is lost in transit, following packets are ignored until the missing packet is retransmitted. For this reason, it is very inefficient. Selective repeat was also slightly better than GBN for higher losses.

Experiment 2 -

1) **Window size (10, 50, 100, 200, 500) for GBN, SR vs Throughput for each GBN, SR, ABT. Loss probability set to 0.2**

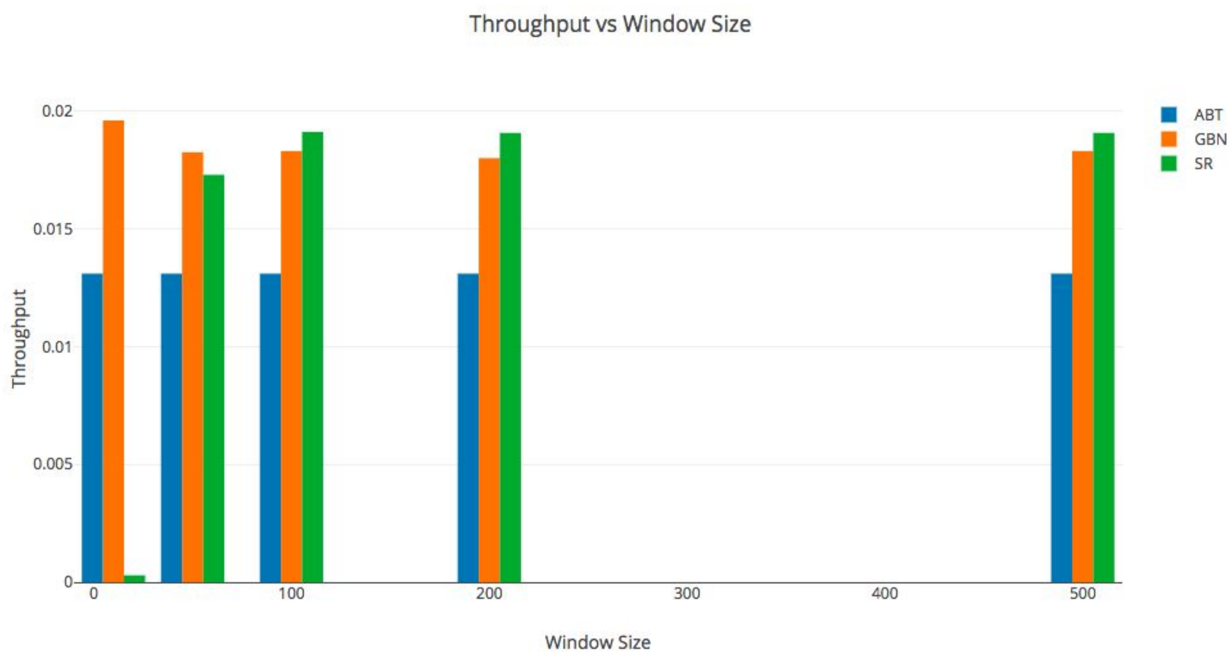
Window Size	ABT	GBN	SR
10	0.0164	0.0199	0.0070
50	0.0164	0.0202	0.020
100	0.0164	0.0202	0.020
200	0.0164	0.0202	0.02006
500	0.0164	0.0202	0.02006



As it can be seen For lower losses, GBN is clearly the winner of all three protocols. Selective repeats performance started to match GBN when window size was increased from 50 onwards. In terms of running time, GBN protocol was worse for higher window sizes. Sometimes taking an hour or so. Selective repeat was quick though. ABT does not depend on window size and as expected its performance is less than GBN and ABT for lower loss.

2) **Window size (10, 50, 100, 200, 500) for GBN, SR vs Throughput for each GBN, SR, ABT. Loss probability set to 0.5.**

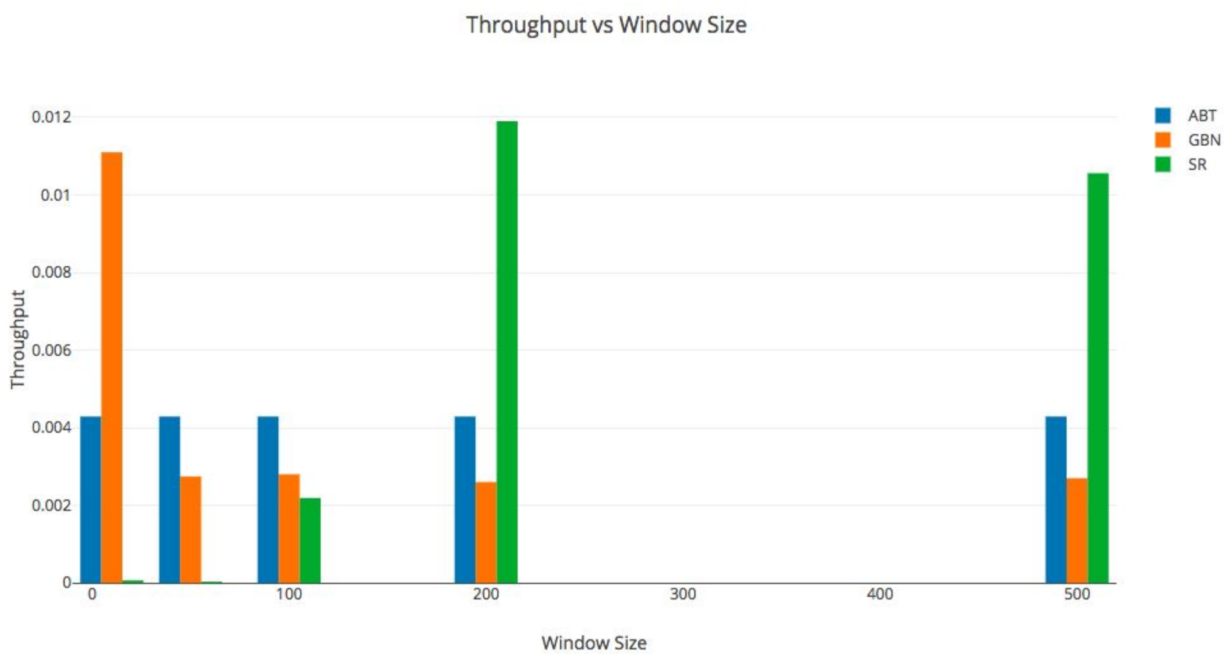
Window Size	**ABT**	**GBN**	**SR**
10	0.0131	0.0196	0.0003
50	0.0131	0.01825	0.01729
100	0.0131	0.0183	0.01911
200	0.0131	0.018	0.01907
500	0.0131	0.0183	0.01907



For a higher loss probability, the advantages of choosing selective repeat seems to appear. As it can be seen that SR performs better than GBN and ABT for larger window sizes.

3) Window size (10, 50, 100, 200, 500) for GBN, SR vs Throughput for each GBN, SR, ABT. Loss probability set to 0.8.

Window Size	ABT	GBN	SR
10	0.00429	0.0111	0.00004
50	0.00429	0.002743	0.00002
100	0.00429	0.0028	0.00219
200	0.00429	0.0026	0.0119
500	0.00429	0.0027	0.01056



Concluding Experiment 2 -

1) As, it can be observed that, if we choose a proper window size, the Selective repeat performs much better than both GBN and ABT in a lossy channel.
2) GBN suffers from performance loss for higher window sizes.
3) Selective repeat performs better at higher window sizes.
4) GBN is very slow for higher window sizes and a lossy channel. If we choose an adaptive timeout scheme, then it’s efficiency can be improved.