PA2-ANALYSIS

CSE-589

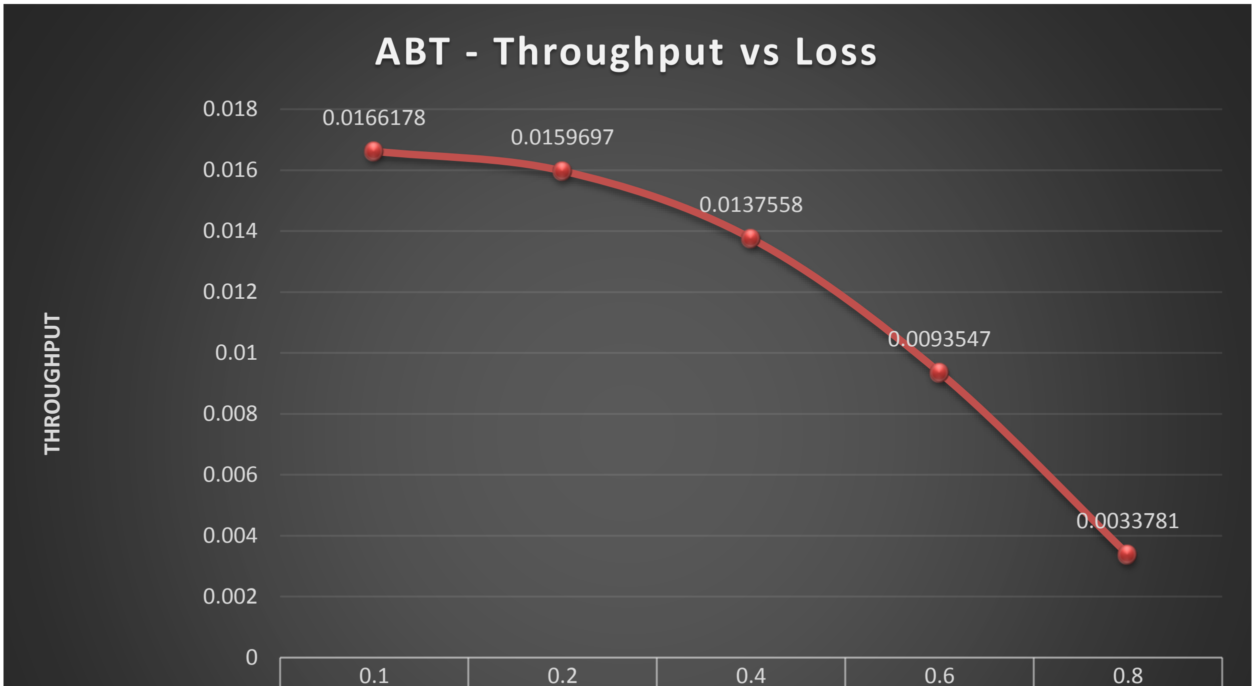
0. “I have read and understood the course academic integrity policy located under this link: http://www.cse.buffalo.edu/faculty/dimitrio/courses/cse4589_f14/index.html#integrity”.
Your submission will NOT be graded without this statement.

1. **Timeout Scheme: -** I have used different timeout schemes for each of the protocols,
   a. ABT: - This protocol used a constant timeout of 8.0 time units. I observed that this constant timeout delivers the best results for all loss rates through experimental analysis.
   b. GBN: - For this protocol I have used different timeouts based on window size. I observed which timeout works best for a particular window size and accordingly arrived at optimal timeout values for all windows required in this assignment.
   c. SR: - As performance of SR greatly depends on timeout values I have used a scheme where the timeout is based on the measured loss rate of packets. The loss rate is maintained as an EMA (exponential moving average) and timeout is updated after every 10 packets sent.

2. **Multiple Timer Scheme: -** I have used a very simple and efficient scheme to implement multiple timer. Every packet sent is assigned a timestamp when it was sent. After that hardware timer is called at unit time to check if any in transit packet has timed out based on its sent time.

3. Experiment 1

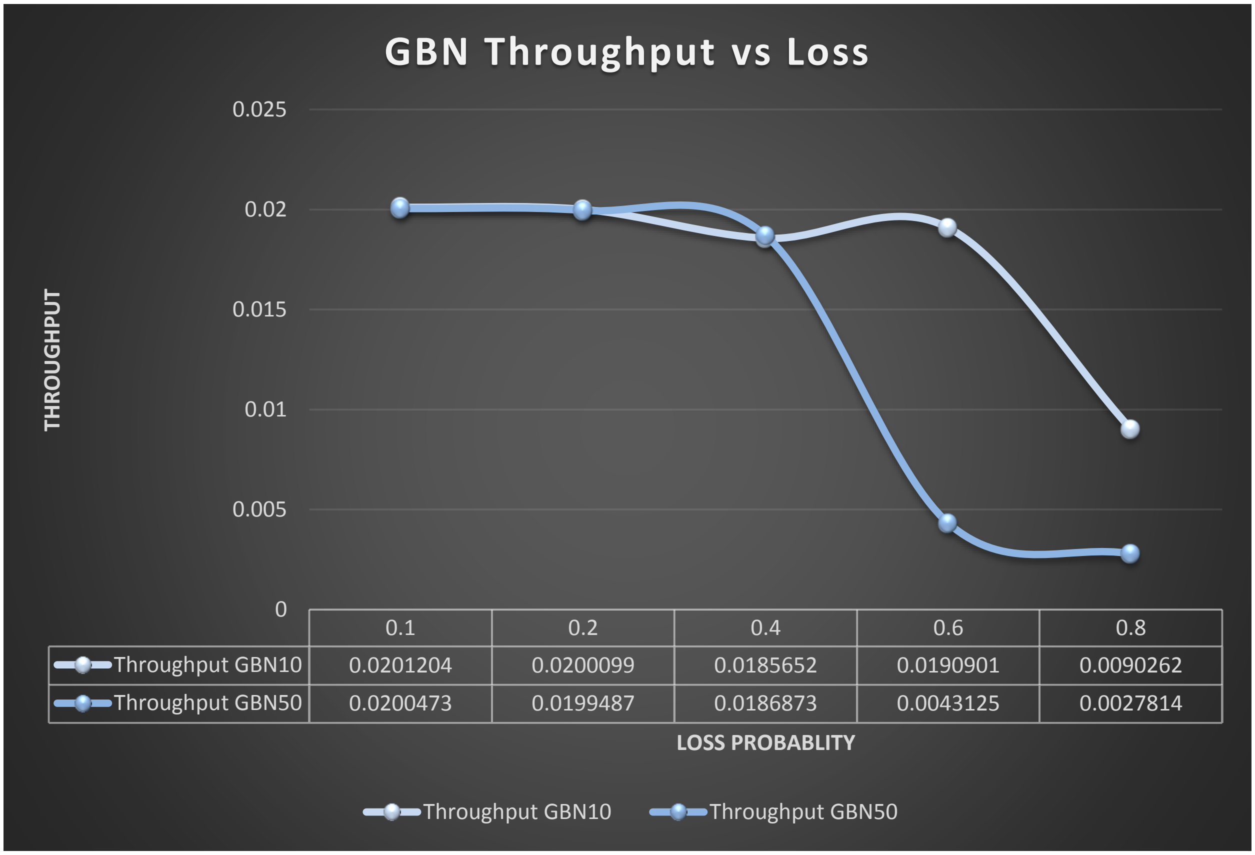
a. Alternating Bit Protocol



	0.1	0.2	0.4	0.6	0.8
Throughput ABT	0.0166178	0.0159697	0.0137558	0.0093547	0.0033781

Observations: - ABT is the least efficient of the 3 protocols implemented and will result in ignoring of packets if the sender has not received acknowledgement for the previously sent packet. As ABT does not buffer any packets on either side the protocol suffers, this is especially relevant when the rate of packet loss and corruption is high. That being said, ABT is a protocol used for learning purpose and hence will not be used in real world scenarios. And as the graph and table show the throughput is inversely proportional to loss rate.

b. Go-Back N

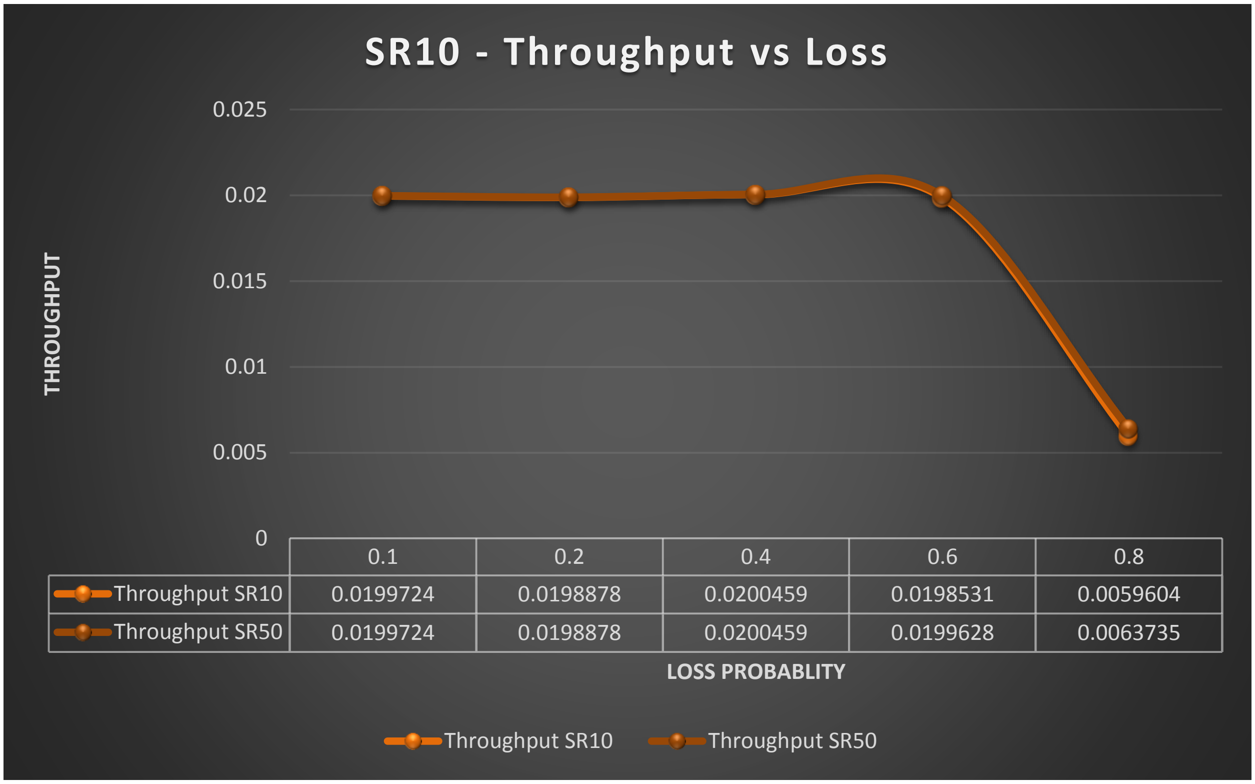


Observations: - GBN protocol suffers because the whole window has to be re-sent even if one loss occurs and also because there is no buffering of packets done at the receiving end. Through experimental data observed I concluded that in GBN a small window size is much better performing as compared to a large window.

An improvement on this protocol could be varying window size according to loss rate; higher loss rate will shrink the window which will result in lower re transmissions of packets and hence will not clog the network. This kind of behavior can be observed in the graph at loss levels 0.6 and 0.8 with window size 50; if the window had been shrunken the throughput could have been much higher as seen in similar loss levels at window size 10.

Another improvement on the FSM implementation would be for the receiver side to buffer packets and then send a cumulative acknowledgement for the received packets, this approach will avoid the discarding of out of order packets on the receiver side.

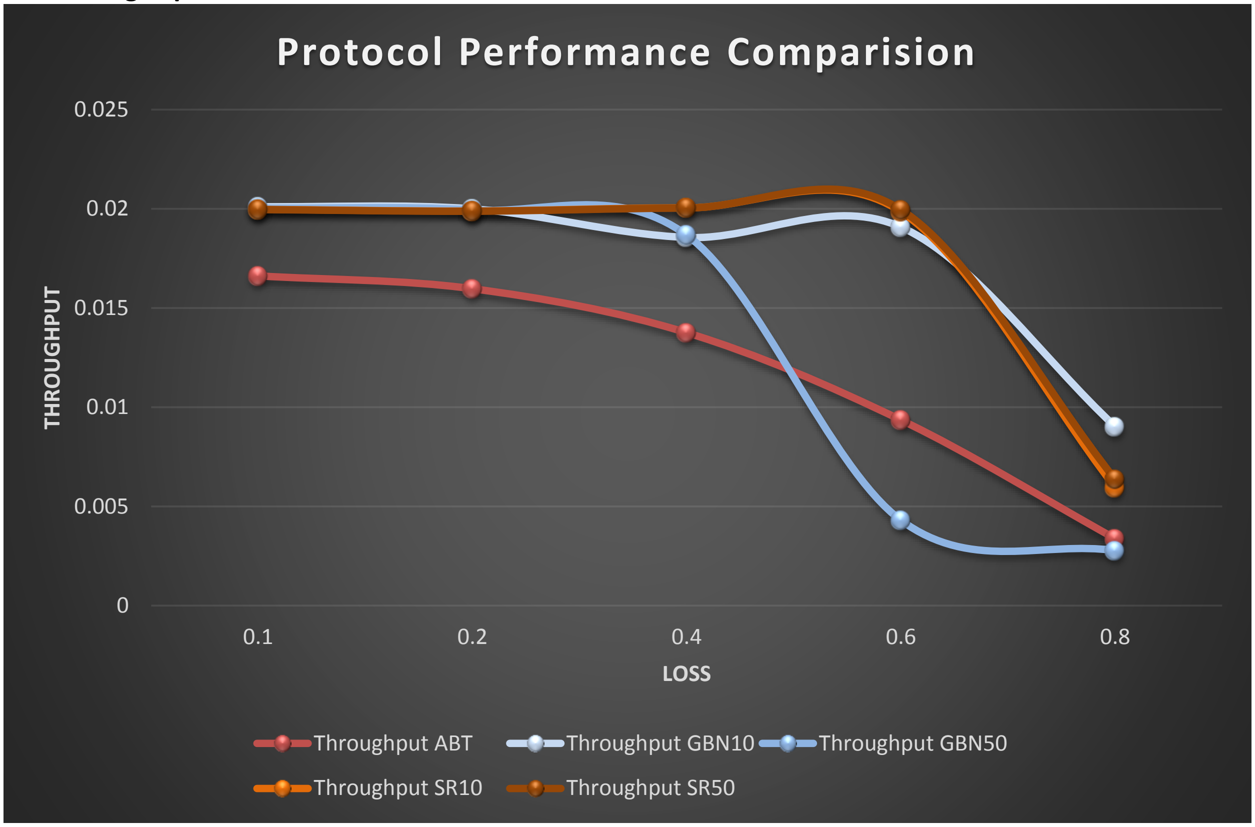
c. Selective Repeat

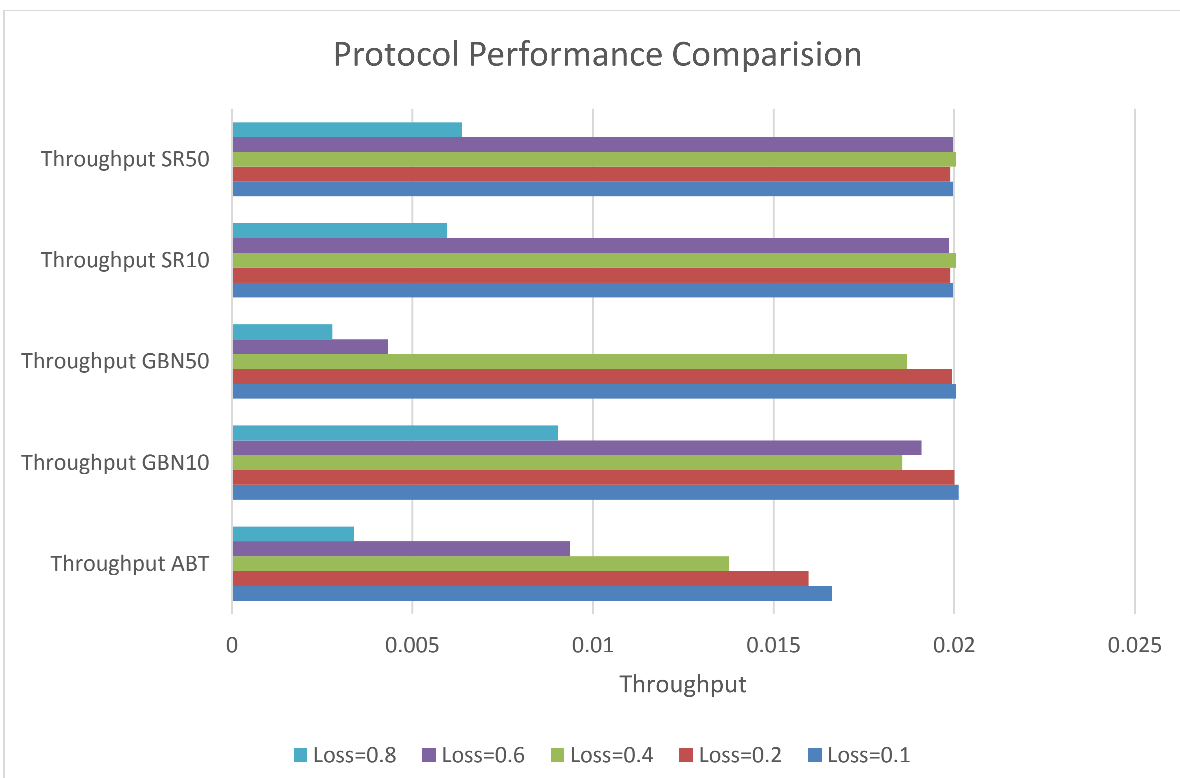


Observations: - Selective Repeat is the most efficient of the 3 protocols implemented and achieved better throughput as compared to ABT and GBN even for loss rates as high as 0.6. As seen in the graph the window size does not play a major part in the throughput but there is a slight throughput increase at loss rate 0.8 for the higher window size.

To further improve the protocol the FSM implementation could be modified at receiver end so that on receiving an already acknowledged packet the receiver will send all the acknowledgments of packets from the sequence number of duplicate packet to the first un-received packet. This approach will flood the network, but could be used in case of higher loss rates and will prevent unnecessary re transmissions from sender. This approach will save bandwidth useful because it's efficient to send an acknowledgement rather than send all the packet again.

d. Concluding Experiment 1





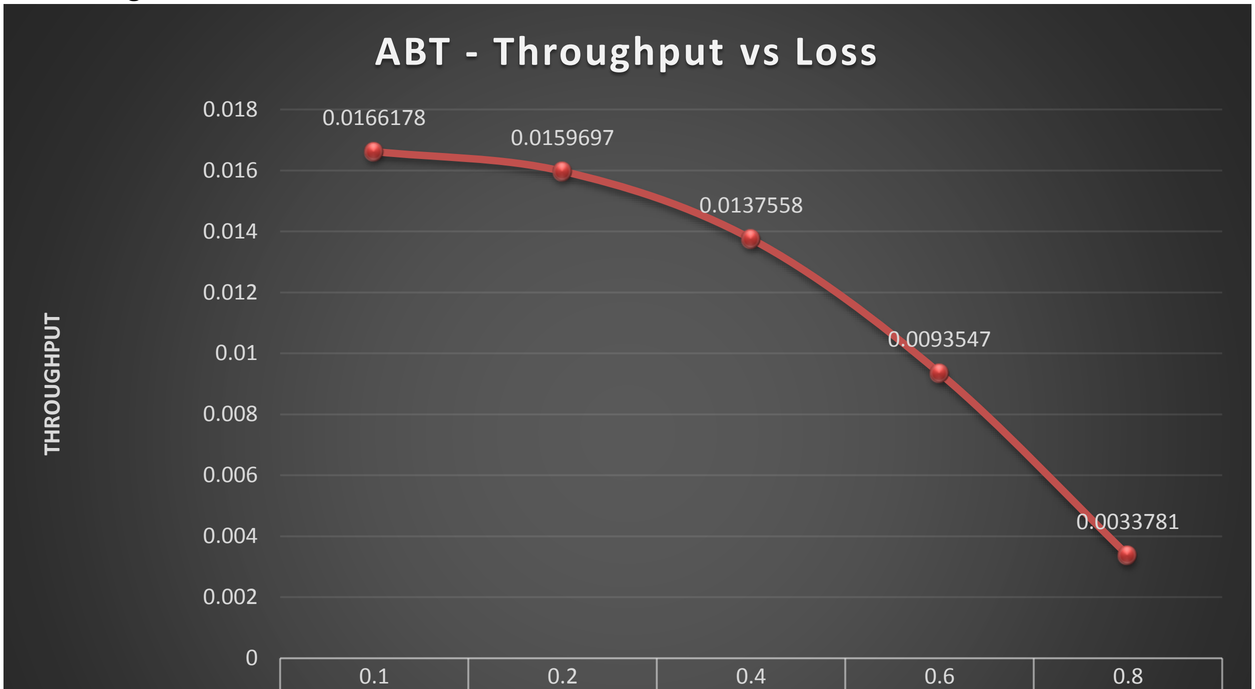
Loss	Throughput ABT	Throughput GBN10	Throughput GBN50	Throughput SR10	Throughput SR50
0.1	0.0166178	0.0201204	0.0200473	0.0199724	0.0199724
0.2	0.0159697	0.0200099	0.0199487	0.0198878	0.0198878
0.4	0.0137558	0.0185652	0.0186873	0.0200459	0.0200459
0.6	0.0093547	0.0190901	0.0043125	0.0198531	0.0199628
0.8	0.0033781	0.0090262	0.0027814	0.0059604	0.0063735

Conclusions: - Selective Repeat is a clear winner as observed from the comparison graphs above. That being said it will be interesting to see comparisons after implementing the improvements suggested for individual protocols.

It can be concluded that in case of GBN it would be beneficial for a higher timeout for larger windows and for window to be shrunken in case of higher loss rates. As far as SR is concerned window size does not matter much as compared to timeout value. The timeout in case of higher loss must be low, although this will cause false timeouts in low loss rates. As far as ABT is concerned, it does not compare favorably with the sliding window protocols even for low loss rates.

4. Experiment 2

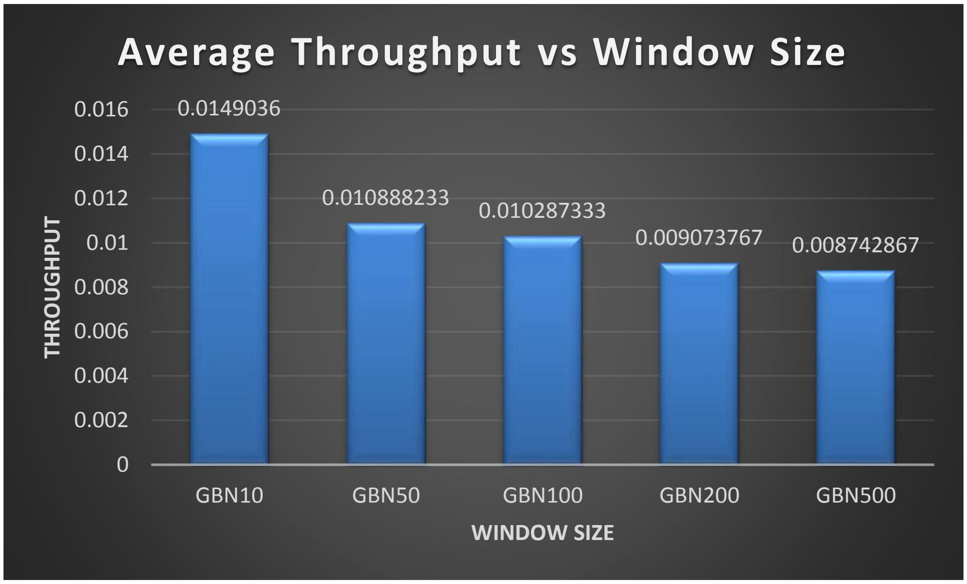
a. Alternating Bit Protocol



	0.1	0.2	0.4	0.6	0.8
Throughput ABT	0.0166178	0.0159697	0.0137558	0.0093547	0.0033781

Observations: -As mentioned ABT does not have a sliding window and it performance is the same.

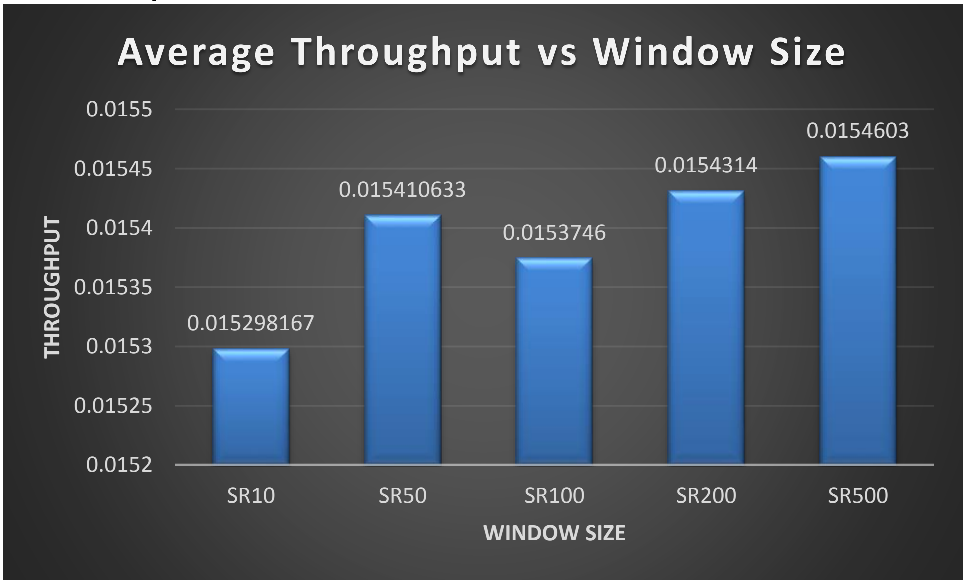
b. Go-Back N



Loss	GBN10	GBN50	GBN100	GBN200	GBN500
0.2	0.0200099	0.0199487	0.0186846	0.0199457	0.019463
0.5	0.0156747	0.0099346	0.0099053	0.0054801	0.0050428
0.8	0.0090262	0.0027814	0.0022721	0.0017955	0.0017228
Average of Window Size	**0.0149036**	**0.010888233**	**0.010287333**	**0.009073767**	**0.008742867**

Observations: - I observed that GBN for a constant timeout value, performs better with a smaller window size. This is because for a single loss the whole window has to be re-sent. Although later I observed varying the timeout according to window size but also loss rate increases performance. With a high window size and high loss rate an increased timeout yields better performance. That being said the best solution for GBN would be varying window size according to loss rate and using a timeout scheme which is inversely proportional to window size.

c. Selective Repeat



Loss	SR10	SR50	SR100	SR200	SR500
0.2	0.0198878	0.0198878	0.0198878	0.0198878	0.0198878
0.5	0.0200463	0.0199706	0.0199706	0.0199706	0.0199706
0.8	0.0059604	0.0063735	0.0062654	0.0064358	0.0065225
Average of Window Size	**0.015298167**	**0.015410633**	**0.0153746**	**0.0154314**	**0.0154603**

Observations: - I observed that in SR a larger window size is yields better performance. Although a larger window will perform better even in higher loss rates there can be additional performance increase when using an adaptive timeout scheme inversely proportional to the loss rate like I have used in my implementation.

d. Concluding Experiment 2

As far as performance goes, there is no beating SR with a high window size which performs better than all other configurations even in high loss rates. That being said it could be interesting to see performance of a GBN with a variable window size. But for now SR wins hand down.

Adaptive timeout is another example where the two protocols differ. In GBN a time out that is directly proportional to window size delivers optimal performance as well as running time, because it will minimize unnecessary re-transmissions. As far as SR is concerned the timeout can be inversely proportional to loss rate as opposed to window size because only the timed out packet is retransmitted as opposed to the whole window.

Concluding this experiment, a larger window size works best for SR and smaller window size will work best for GBN.