CSE 489/589

Programming Assignment 2 Report

RELIABLE DATA TRANSFER PROTOCOLS

I (We) have read and understood the course's academic integrity policy.

Group and Contribution:

- Name: Pinnam Sampath Bhargav • UB ID/ Person Number is: 50419162
  - Email: sampathb@buffalo.edu
  - Contribution: code, report, design.
- Name: Motamarri Soma Naga Venkata Sai Anvesh • UB ID/ Person Number is: 50414978
  - Email: somanaga@buffalo.edu
  - Contribution: code, report, design.
- Name: Chennakesava Sukesh Myla • UB ID/ Person Number is: 50413326
  - Email: cmyla@buffalo.edu
  - Contribution: code, report, design.

ALTERNATING BIT PROTOCOL:

It is the most basic way of flow control. The sender will send one frame at a time to the receiver in this protocol. The sender will come to a halt and await the receiver's acknowledgement. This period (i.e., the time between sending a message and receiving an acknowledgement) is the sender's waiting time, and the sender is completely idle during this time. When the sender receives an acknowledgment (ACK), it will transmit the next data packet to the receiver and wait for another acknowledgment, and so on, as long as the sender has data to send.

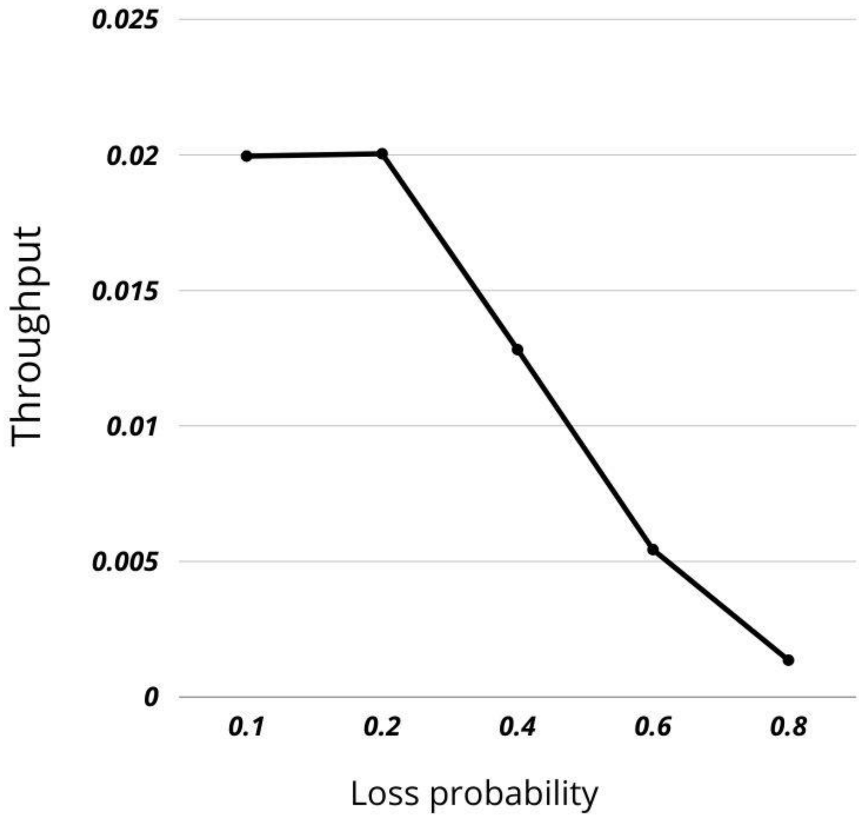
TIMEOUT SCHEME:

A constant time has been used.

Timeout = 20

THROUGHPUT ANALYSIS: The figure shows the throughput of Alternating Bit protocol for different loss probabilities.

Loss Probability	**Alternating bit protocol**
0.1	0.0199543
0.2	0.0200457
0.4	0.0128183
0.6	0.005439
0.8	0.0013576



<u>GO-BACK N:</u>

In GO-BACK-N, the sender controls the flow of packets, meaning the sender is allowed to transmit multiple packets without waiting for acknowledgement. A timer is maintained to keep track of the oldest unacknowledged packet, as the receiver only sends the cumulative acknowledgements.

TIMEOUT SCHEME:

The timer value was determined after a lot of experiments with different timeout values to find the best possible timeout value as other timeout values keep the program running for a longer time.

Timeout = 20

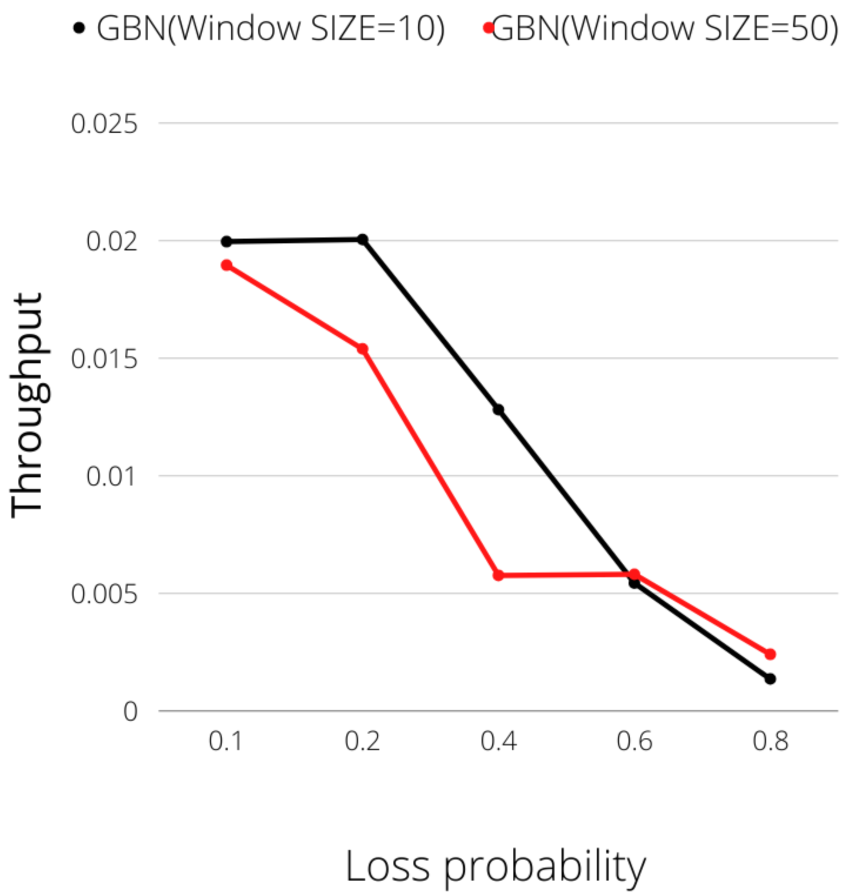
THROUGHPUT ANALYSIS:

The results for window sizes 10 and 50 are shown in the given chart.

We can see that with increase in the window size the throughput decreases and with the increase in the loss probability the given throughput decreases.

Loss probability	**GBN (10)**	**GBN (50)**
0.1	0.018934	0.018954

0.2	0.0149643	0.015395
0.4	0.0134122	0.005766
0.6	0.002969	0.005812
0.8	0.000473	0.002405



<u>SELECTIVE REPEAT</u>:

The SELECTIVE REPEAT is a sliding window protocol like GO-BACK-N.

In GO-BACK-N the drawback is that even a single packet loss or error can lead to a lot of retransmissions this is avoided in selective repeat.

TIMEOUT SCHEME:

For Selective repeat, we maintain timers for each individual packet (unlike in GBN where the timer is only associated with the current window base), and we have only a single hardware timer which has been used to mimic the operation of multiple logical timers.

Timeout = 20

Multiple Software timers

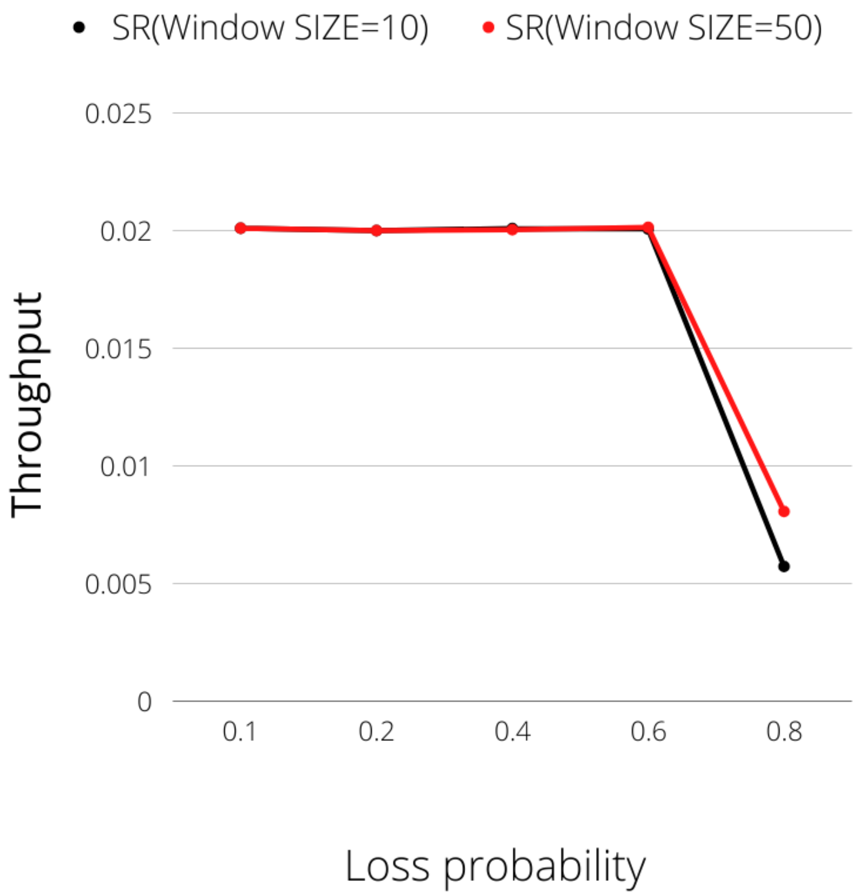
The goal was to keep the timer values in the array in relation to one another. When a packet is transmitted, the array is updated with the value of (current time + timer value). When a timer interrupt occurs, the timer value in the array that has expired is increased by the original timer value units, and the next timer value to begin the timer is computed. The next timer value is determined by locating the array's minimum value and subtracting the current time from it. We can assure that each packet is retransmitted when its timer ends this way.

THROUGHPUT ANALYSIS:

By observing the given chart, we can say that with increase in the window size the throughput is increases slightly.

The results for window size 10 and 50 are as follows:

LOSS PROBABILITY	10	50
0.1	0.020095	0.020095
0.2	0.0199946	0.019995
0.4	0.020088	0.020034
0.6	0.020068	0.020137
0.8	0.005723	0.008058



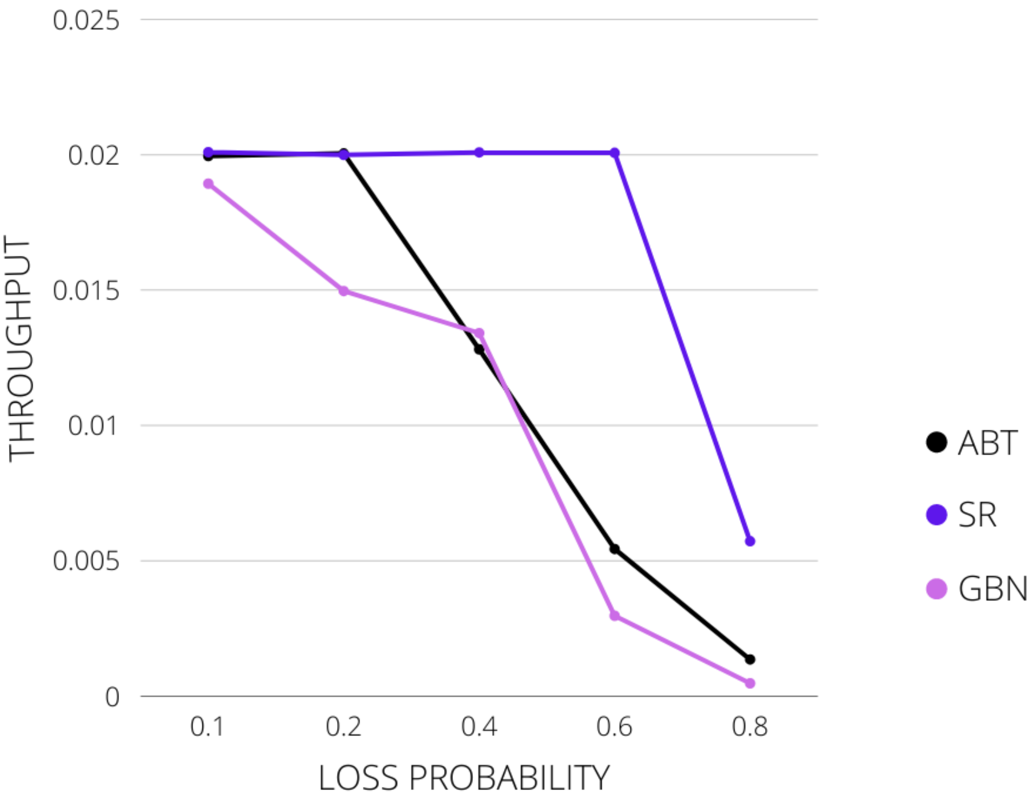
Experiment 1

With loss probabilities: {0.1, 0.2, 0.4, 0.6, 0.8}, compare the 3 protocols' throughputs at the application layer of receiver B. Use 2 window sizes: {10, 50} for the Go-Back-N version and the Selective-Repeat Version

A. For window size 10:

The observed results are:

Loss Probability	ABT	GBN	SR
0.1	0.0199543	0.018934	0.020095
0.2	0.0200457	0.0149643	0.0199946
0.4	0.012813	0.0134122	0.020088
0.6	0.005439	0.002969	0.020068
0.8	0.0013576	0.000473	0.005723



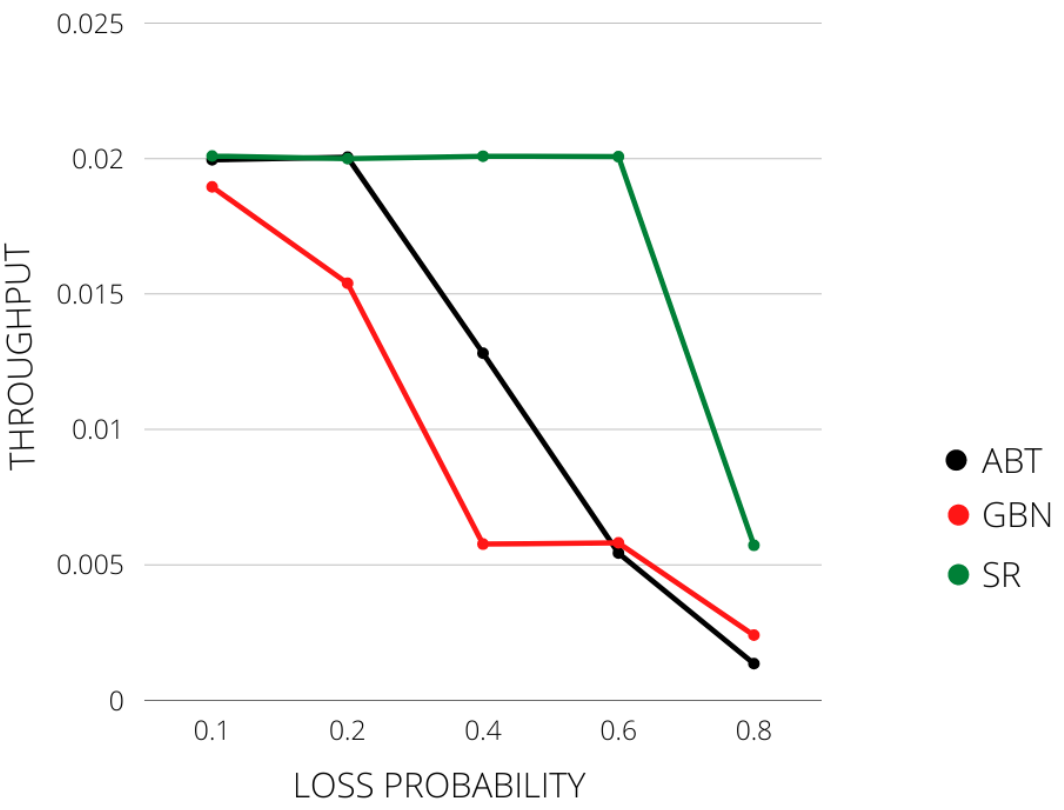
Observation:

- The SR and ABT has almost the same throughput when the loss probability when loss is 0.1 or 0.2 and GBN is better than ABT for loss probability 0.4.
- At higher loss probability the SR is performing better.

B. For window size 50:

The observed results are

Loss Probability	ABT	GBN	SR
0.1	0.0199543	0.018954	0.020095
0.2	0.0200457	0.016757	0.019995
0.4	0.012813	0.005766	0.020034
0.6	0.005439	0.005812	0.020137
0.8	0.0013576	0.002405	0.008058



Conclusion:

From experiment 1 we can say the following:

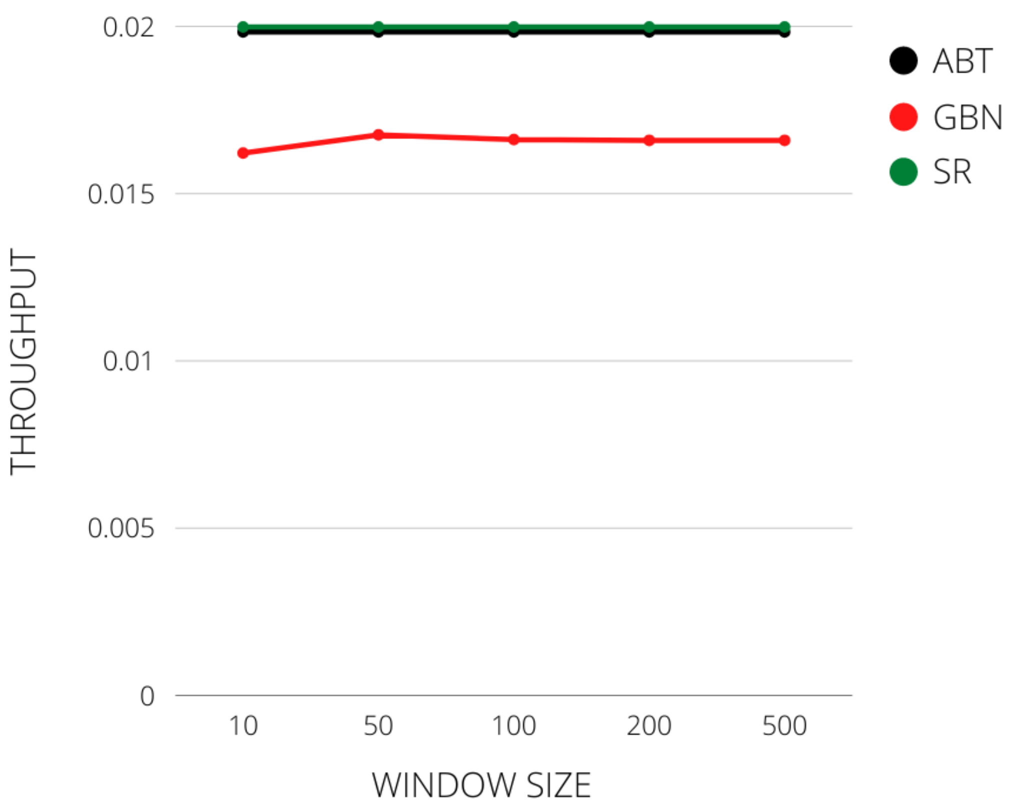
- SR is most efficient out the three protocols.
- The throughput of GBN decreases with increase in window size and loss probability.
- ABT throughput decreases as the loss probability increases.
- Since the timer implemented is not adaptive we can see the degradation of GBN's performance along with an increase in window size. Better implementation of timer might give some good results.

Experiment 2

With window sizes: {10, 50, 100, 200, 500} for GBN and SR, compare the 3 protocols' throughputs at the application layer of receiver B. Use 3 loss probabilities: {0.2, 0.5, 0.8} for all 3 protocols.

1) **Loss probability:**0.2

WINDOW SIZE	ABT	GBN	SR
10	0.020046	0.016214	0.0199946
50	0.020046	0.016757	0.019995
100	0.020046	0.016617	0.019995
200	0.020046	0.016593	0.019995
500	0.020046	0.016593	0.0199946

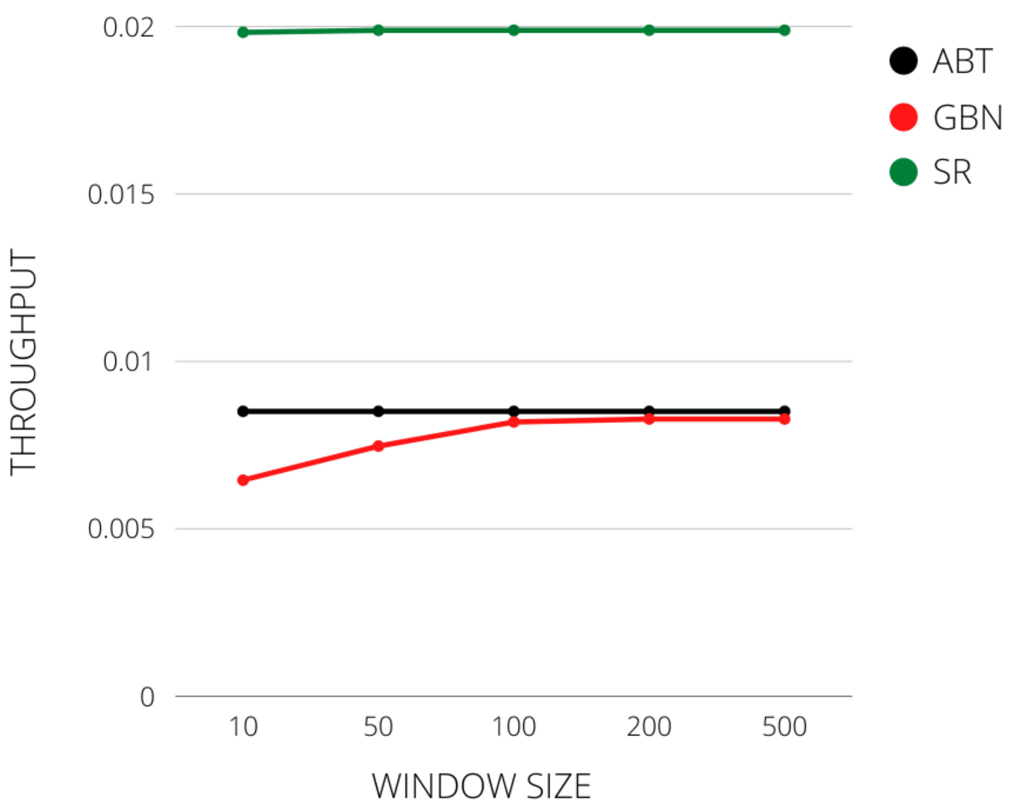


Observation:

1. As we can see from the above chart, we can say that SR and ABT are almost Similar when the loss probability is 0.2 and as we can see it is unaffected by the changes in the window size.
2. In case of GBN the throughput is less when we compare it with GBN and SR.
3. So, for the loss probability 0.2, SR and ABT both perform good when compared with GBN.

2) Loss probability:0.5

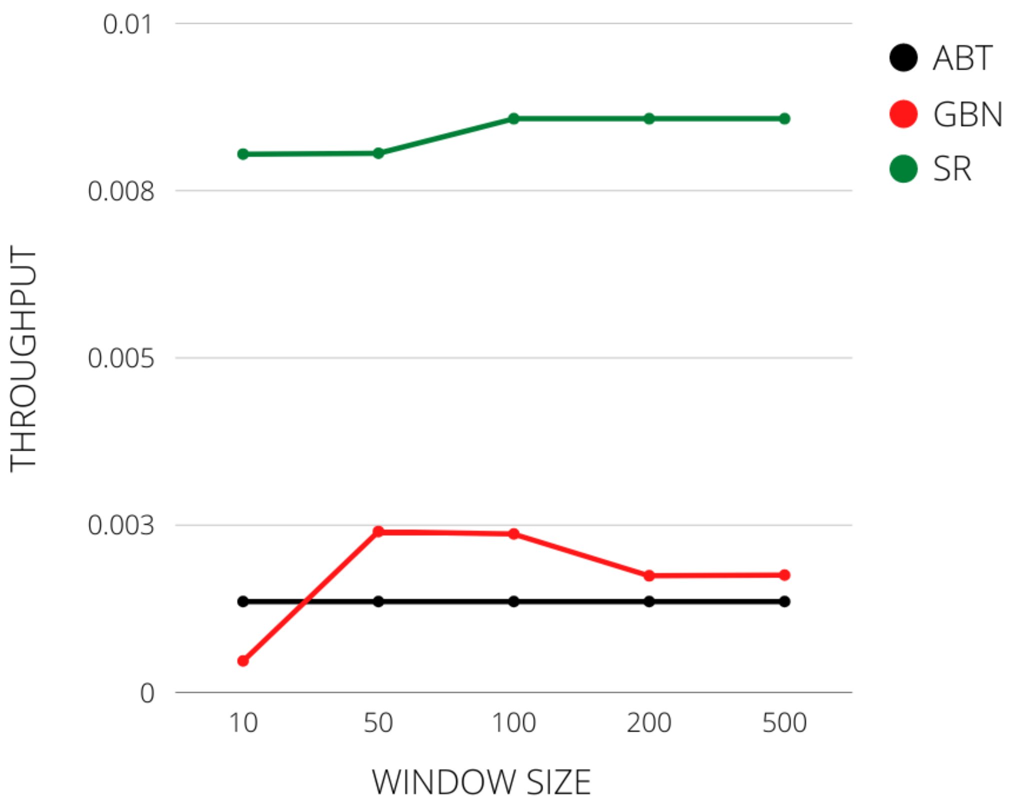
WINDOW SIZE	ABT	GBN	SR
10	0.008508	0.006451	0.019821
50	0.008508	0.007470	0.0198838
100	0.008508	0.008190	0.019884
200	0.008508	0.008275	0.019884
500	0.008508	0.008275	0.019884



Observation:
From the above chart we can observe that among the three other reliable transport protocols the SR throughput is better compared with the other reliable data transfer protocols.

3) **Loss probability**:0.8

WINDOW SIZE	ABT	GBN	SR
10	0.001358	0.000469	0.008046
50	0.001358	0.002405	0.008058
100	0.001358	0.002367	0.008574
200	0.001358	0.001742	0.008574
500	0.001358	0.001753	0.008574



Observation:

We can clearly see from the chart that the given throughput is higher for SR when compared with the others as the throughput isn't unaffected with the window size.

Conclusion:

1. From the above charts we can clearly see that the SR performs best among the three reliable protocols in all cases.
2. ABT is more efficient when the loss probability is less and isn't affected with the increase in window size.
3. In SR the Retransmission happens only for the timeout packet and hence there is a chance for the throughput to increase momentarily as loss increases.

4. It’s clear that the SR Performs better and GBN throughput decreases as the loss probability increases and the window size increases.