Programming Assignment 2 - Report

Prepared By:
Vijaya Harshavardhan Palla
50246589

I have read and understood the course academic integrity policy

Table Of Contents

SL.No.	Title	Page No.
1.	Timeout Scheme for three protocols	3
2.	Multiple Timers Implementation of Selective Repeat Protocol	7
3.	Performance comparison - Experiment 1	8
4.	Performance comparison - Experiment 2	12

1.Timeout Scheme for three protocols

1.1 Alternating Bit Protocol (rdt 3.0)

The timeout for alternating bit protocol has been chosen by considering the following points:

- Timer is started for every packet that is sent
- The possibility of congestion is negligible as it is a stop and wait protocol and not many packets will be there in the network at any given time
- Sender rate is greater than the receiver rate

Timer Value for ABT (here onwards ABT will be used in place of Alternating Bit Protocol) has been selected based on the above stated points.

Since congestion in the network is very negligible, it is very important to make use of this factor and select a timer which is suitable to send packets without much delay.

Hence following scheme has been used to find an appropriate timer which has a good throughput but without creating an overload on the network by excessive retransmission of packets.

- **Function get_sim_time()** is used to calculate the time at which each packet is sent and the time at which acknowledgement for each packet is received. (Table 1)
- Average round trip time is calculated based on the values obtained
- Obtained average round trip time is chosen as the ideal timer value

To prove that the chosen value is ideal, throughput values of various scenarios for timer values less than and greater than the chosen value have been calculated and tabulated (Table 2)

Packet No. #	Simulation Send time	Simulation Recv. Time	RTT
1	72.3351	82.037	9.7019
2	107.972	119.815	11.843
3	144.767	155.834	11.067
4	210.621	227.053	16.432
5	272.465	286.597	14.132
6	361.929	367.492	5.563
7	434.569	441.249	6.68
8	504.304	518.039	13.735
9	586.915	599.403	12.488
10	657.68	669.335	11.655

11	694.226	707.959	13.733
12	713.921	727.336	13.415
13	744.059	758.264	14.205
14	793.867	805.18	11.313
15	891.352	906.685	15.333
16	941.857	955.493	13.636
17	972.727	987.464	14.737
18	1026.99	1045.16	18.17

Table 1

Average RTT: 12.65771667

Sl.No	Timer Value (Time units)	Throughput (packets/time units)
1	11	0.01277
2	12.6	**0.01655**
3	15	0.0153

Table 2

The throughput values of above table are formulated by taking the average of throughput when loss is 0.0, 0.2, 0.4.

From the table it can be verified that the chosen timer is an ideal timer for the implemented ABT protocol.

1.2 Go-Back-N Protocol

After careful consideration of the following facts an ideal timer value can be chosen

- All the packets in the window are sent in parallel (pipeline protocol)
- All the packets in the window are retransmitted when timeout occurs for that particular window

Based on the above facts, following statements about timer value can be concluded

- Timer value should be sufficient enough for all the packets to be sent and for all the packets to receive acknowledgement
- Timer value should not be too large because in case of loss or corruption if timer value is too large it takes too long before sender realizes and retransmits the entire window

There is heavy congestion when packets are corrupted in the network as an entire window needs to be retransmitted when there is corruption. This induces congestion.

From experiment it has been observed that congestion is very high when corruption is >= 0.6. So let us consider the case when **corruption is 0.4** and the **window size is 10** find throughput values for various timers.
Starting observations from a timer value which is greater than that of ABT.
For simplicity sake, choose 20 as first value and increment it by 10 units and tabulate the results.

Timer Value	**Throughput(Corruption 0.6)**
20	0.002442
30	0.001910
40	0.002571
50	0.01187
60	0.010232

Table 3

From the above table, it can be safely concluded that the throughput value is highest when timer value is 50 time units. Hence **50** time units is used as a static timer value when window size is **10** for GBN protocol.
But this value changes when window value changes and it has been found that when window is 20 the timer value required is 100. So, now it becomes a necessity to find a pattern between ideal timer values and the window size.
Tabulating the ideal timers for various window sizes below to find the pattern.

Window Size	**Ideal Timer value (In Time units)**
10	50
20	100
30	150
40	200

Table 4

Based on the above table it can be observed that timer value is a five times the window size and hence timeout scheme chosen for GBN is **5*WinSize.**

1.3 Selective Repeat Protocol (SR)

Facts to consider before designing a timeout scheme for SR protocol:

- All the packets present in the window are sent in parallel unlike ABT

- Each packet has a timer and a timeout value unlike in GBN protocol
- Only the packet whose acknowledgment is received or corrupted will be retransmitted unlike in GBN protocol

Considering above facts and also one another important fact that during corruption there will be high congestion.
Hence it is very important to determine a timer which does not create congestion in the network by excessive retransmission of data.

Finding the average round trip time taken by each packet when the corruption is 0.6 (This value is chosen as it is observed that there is a heavy congestion during this period).
The integer value of this average has come to 30 time units.
Tabulating the throughput values for various timer and verifying if the average value can be chosen as the timer value.

Timer value	**Throughput (corruption 0.6)**
15	0.0042
30	0.02
50	0.018

Table 5

From the table, timer value of **30** has been chosen as an ideal and static timer.

2. Multiple Timers Implementation of Selective Repeat Protocol

2.1 Assumptions

Following assumptions have been made to implement multiple timers logic for SR Protocol

a. The timeout value is static for the entire transmission
b. If a hardware timer is started then it will not be stopped. It waits until timer expires and timer interrupt is called
c. get_sim_time() function gives accurate present simulation time

2.2 Variables used

timers[]: Array used to store the values at which each packet is sent
timerPointer: A variable which points to the array element in timers array at which the timer is started
timer: A variable used to store default timeout value

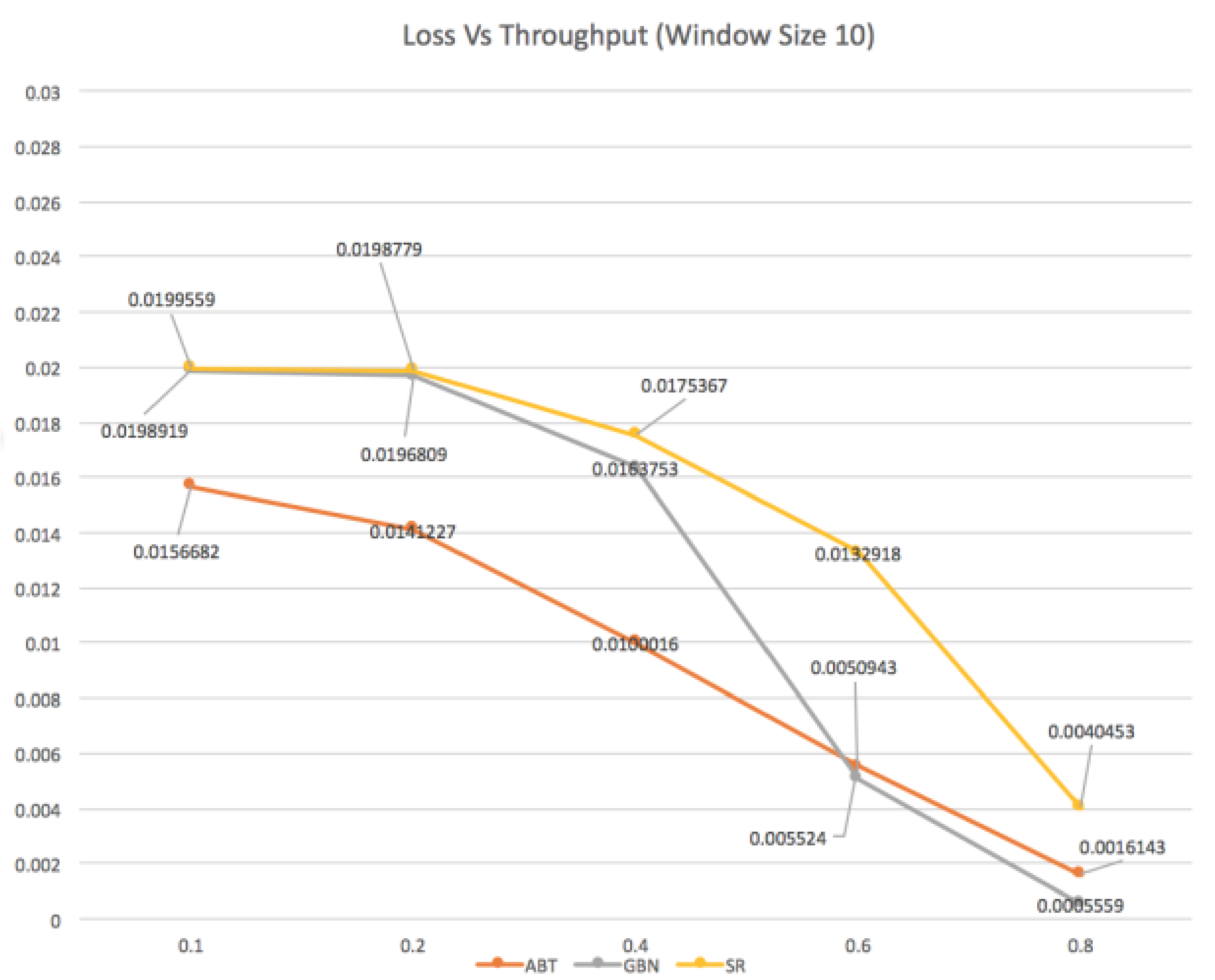
2.3 Implementation

a. Before sending a packet to layer 3, the time at which the packet is sent is added into the timers array
b. If no timer is started then a timer is started with a default value and timerPointer is initialized pointing to the array element for which the timer has started
c. When timer interrupt occurs, increment the timerPointer value and obtain the time at which the next packet has been sent
d. Add default timer value to the obtained value
e. Subtract current simulation time from the above result and we get a relative timer value for the time which next packet should be run before timer interrupt occurs
f. Use the above result as a parameter in startTimer function

3. Performance Comparison

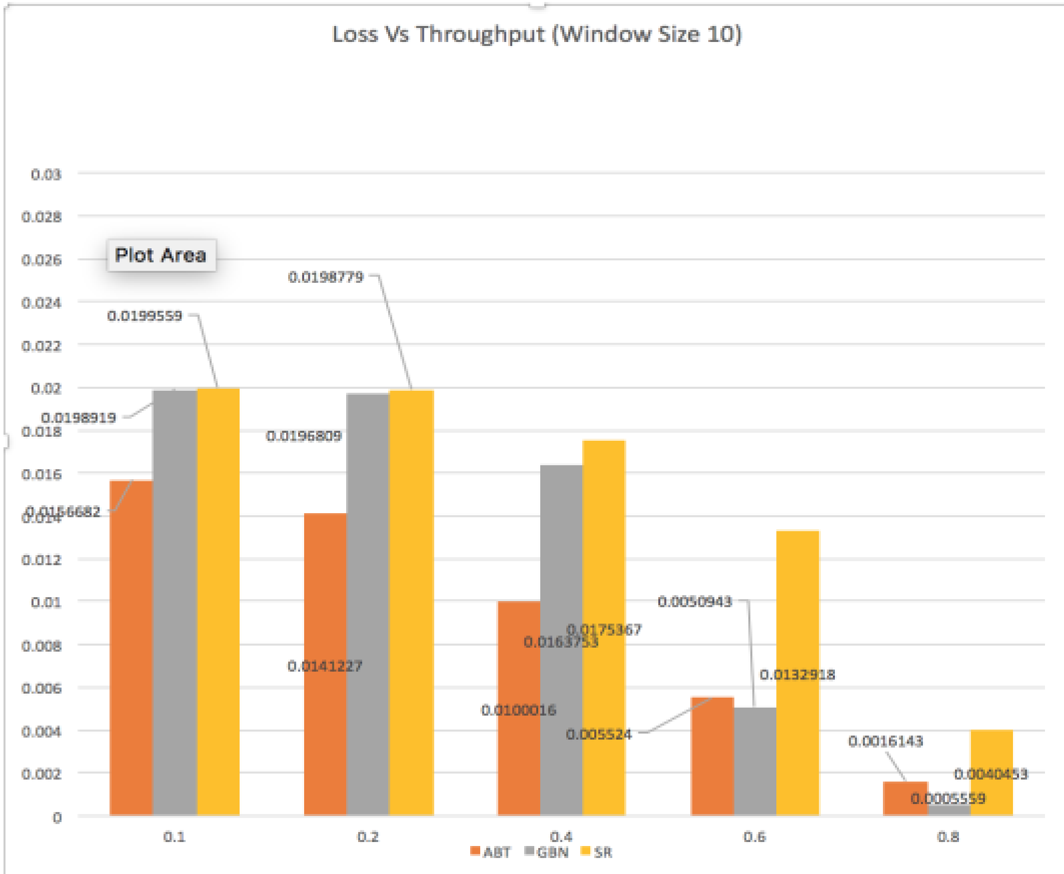
Experiment 1

3.1 Graphs For Window Size 10



Graph 1

X Axis : Loss Probability ; Y Axis : Throughput Value



Graph 2

X Axis : Loss Probability ; Y Axis : Throughput Value

General Observations (Window Size = 10)

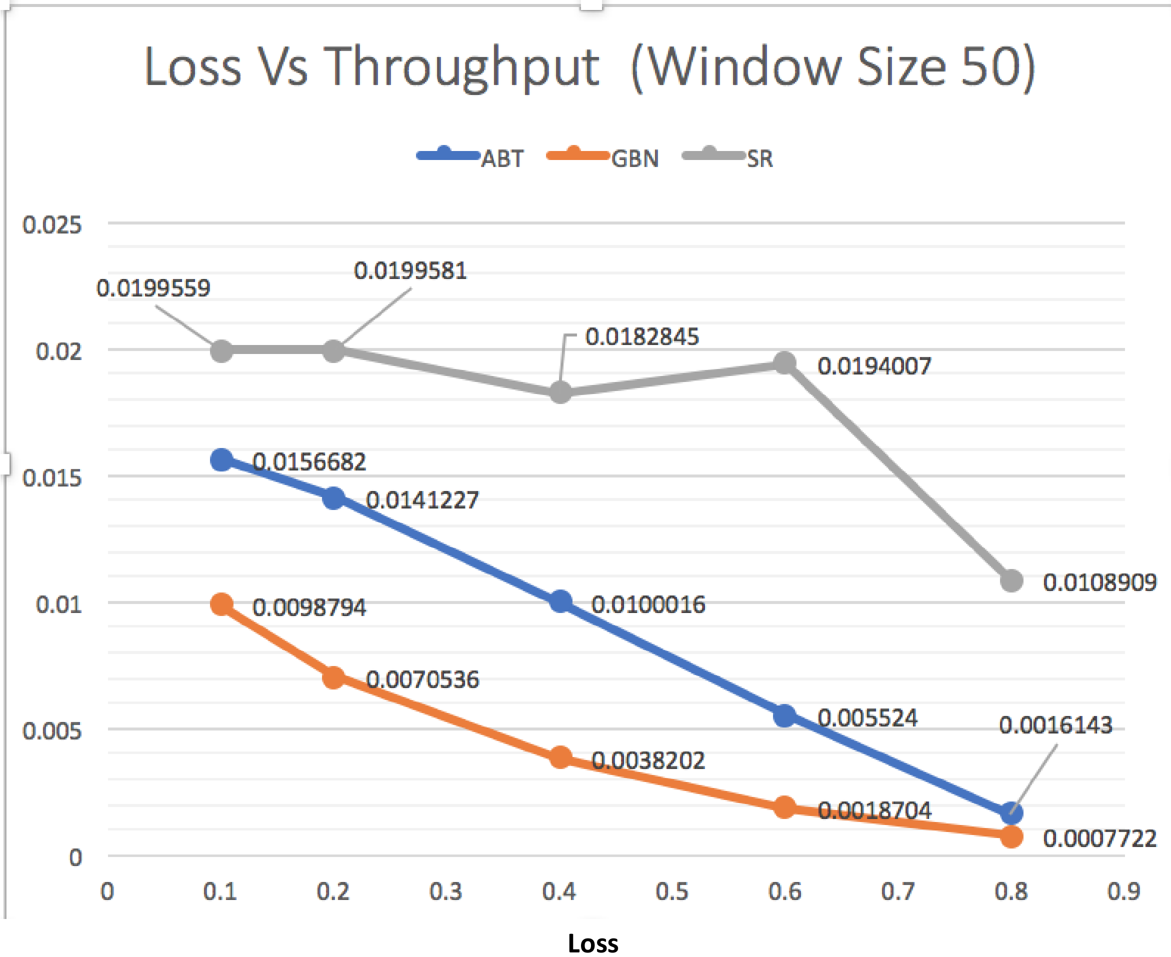
a. From the above bar chart it is clear that SR has a better throughput all through.
b. GBN maintains the same throughput as SR till loss of 0.4 but drops down after that. This behavior can be attributed to the retransmission of entire window concept used in GBN. When there is a huge loss, GBN needs to retransmit the entire window which results in less throughput unlike SR which retransmits lost packets alone.
c. It can also be inferred from the above two graphs that ABT protocol has a better throughput when loss is high justifying the expected behavior.

From above observations it can be safely concluded that GBN protocol is not preferable in a network which has high loss probability.

Expected Behavior :

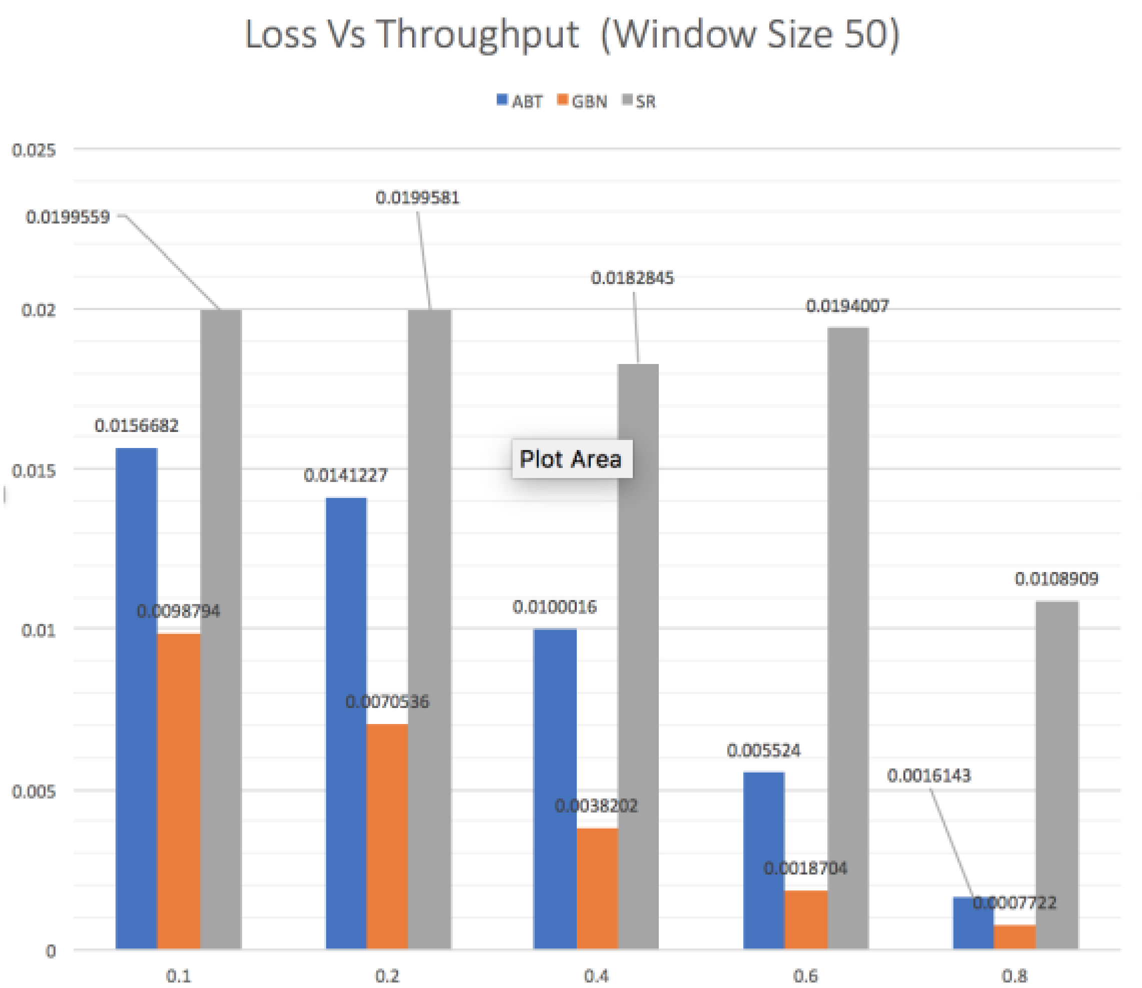
a. ABT meets the expected behavior of having lesser throughput than the other two protocols when loss is less
b. ABT may have higher throughput than GBN when loss is high and same thing is noticed in the above graph

3.2 Graphs For Window Size 50



Graph 3

X Axis : Loss Probability ; Y Axis : Throughput Value



Graph 4

X Axis : Loss Probability ; Y Axis : Throughput Value

General Observations (Window Size = 50)

a. Clearly SR dominates again by maintaining a good and decent throughput for all the loss probabilities
b. But GBN seems to have dropped down drastically when window size is increased.
c. ABT has better throughput than GBN for every loss probability

Expected Behavior/ Variations

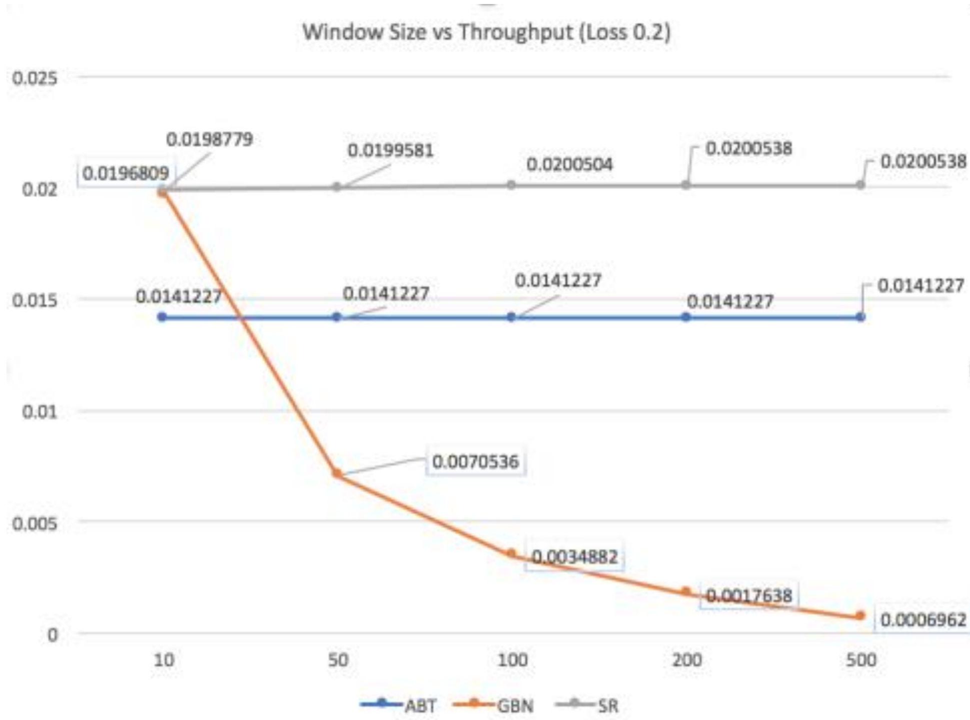
a. GBN throughput decreased when window size is decreased owing to retransmission of a bigger window when loss or corruption of packet occurs
b. GBN has lesser throughput than ABT for every loss probability which is an anomaly because GBN is expected to perform well when loss probability is low

Conclusion

a. SR is effective protocol when window size of 50 is chosen and loss is high
b. If it comes to either ABT or GBN then based on the above observations it is safe to go with ABT protocol

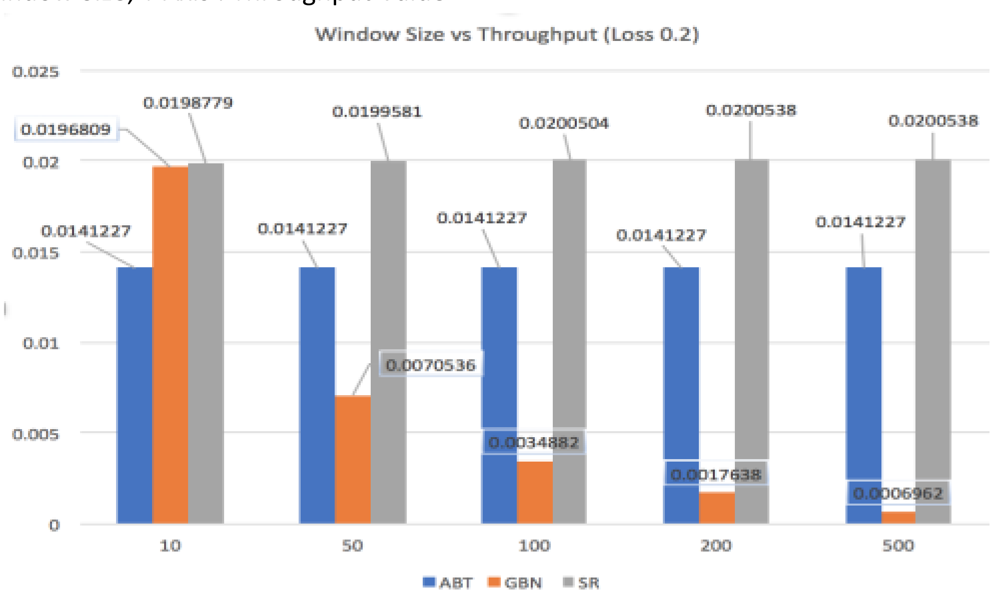
4. Performance Comparison
Experiment 2

4.1 Graphs For Loss probability 0.2



Graph 5

X Axis : Window Size; Y Axis : Throughput Value



Graph 6

X Axis : Window Size; Y Axis : Throughput Value

General Observations (Loss 0.2)

a. ABT throughput remains constant for various window sizes as it does not depend upon window size
b. GBN throughput is decreasing drastically upon increasing the window size
c. SR maintains steady and good output for various window sizes

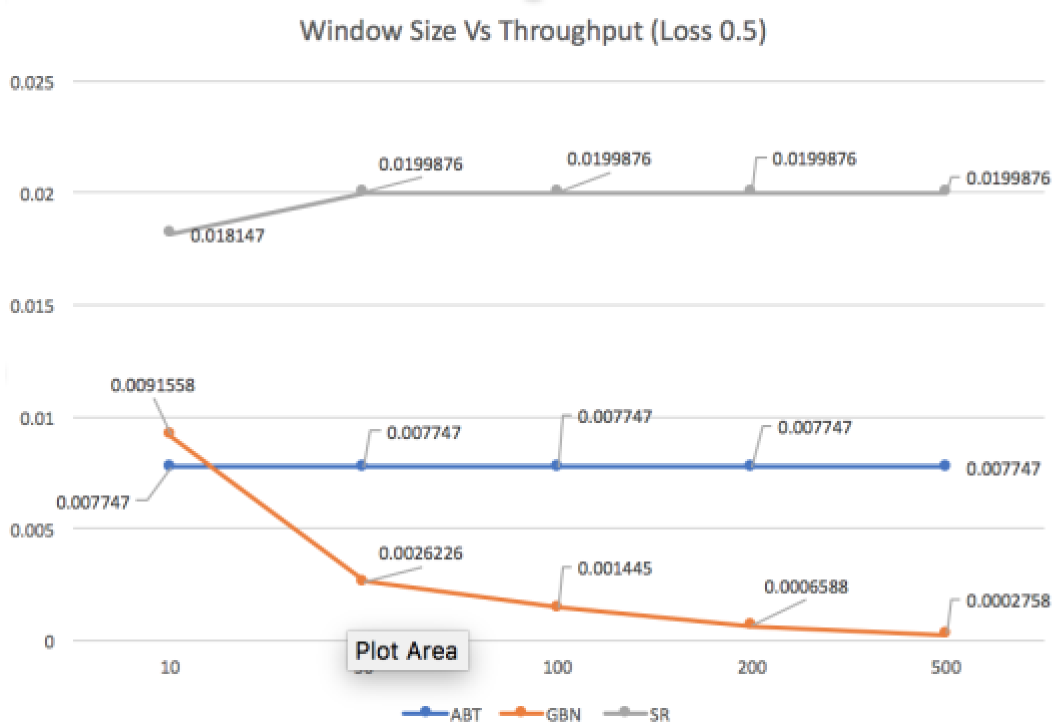
Expected Behavior/Variations

a. ABT is maintaining constant throughput as expected
b. SR has a better throughput and almost constant output for various window sizes as expected
c. GBN has a decreasing throughput upon increasing window size, this is because of retransmission of entire window when there is a loss

Conclusion

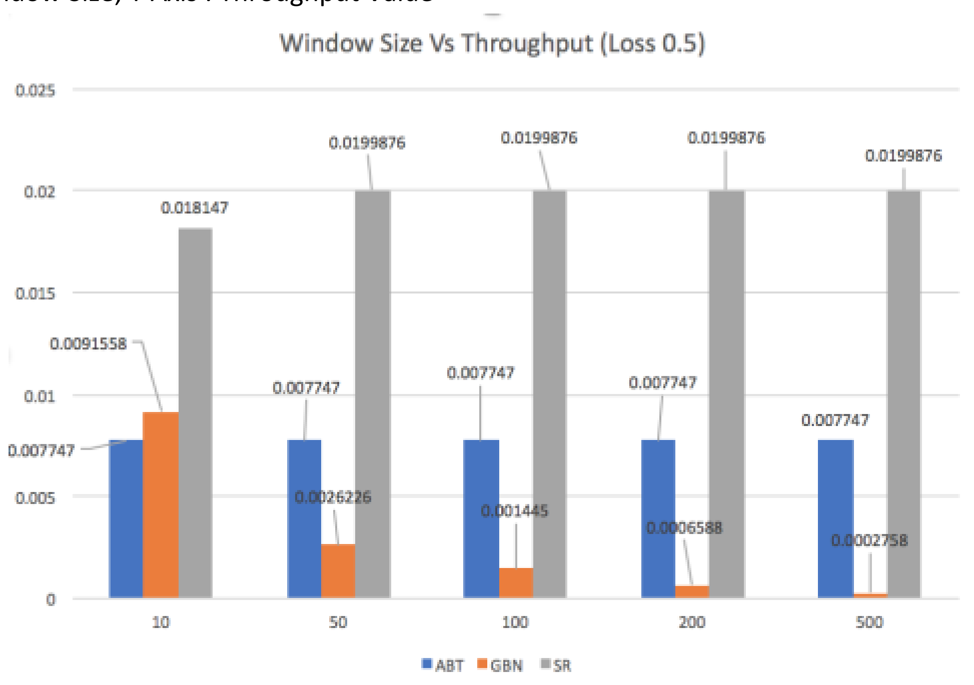
a. GBN is preferable when loss probability is very less and window size is small although SR has slightly better throughput than GBN. This is because of the implementation cost and buffers which are required in the receiver side
b. SR is preferable if bandwidth is of high priority

4.2 Graphs For Loss probability 0.5



Graph 7

X Axis : Window Size; Y Axis : Throughput Value



Graph 8

X Axis : Window Size; Y Axis : Throughput Value

General Observations (Loss 0.5)

a. ABT throughput remains constant for various window sizes as it does not depend upon window size
b. GBN throughput is decreasing drastically upon increasing the window size
c. SR maintains steady and good output for various window sizes

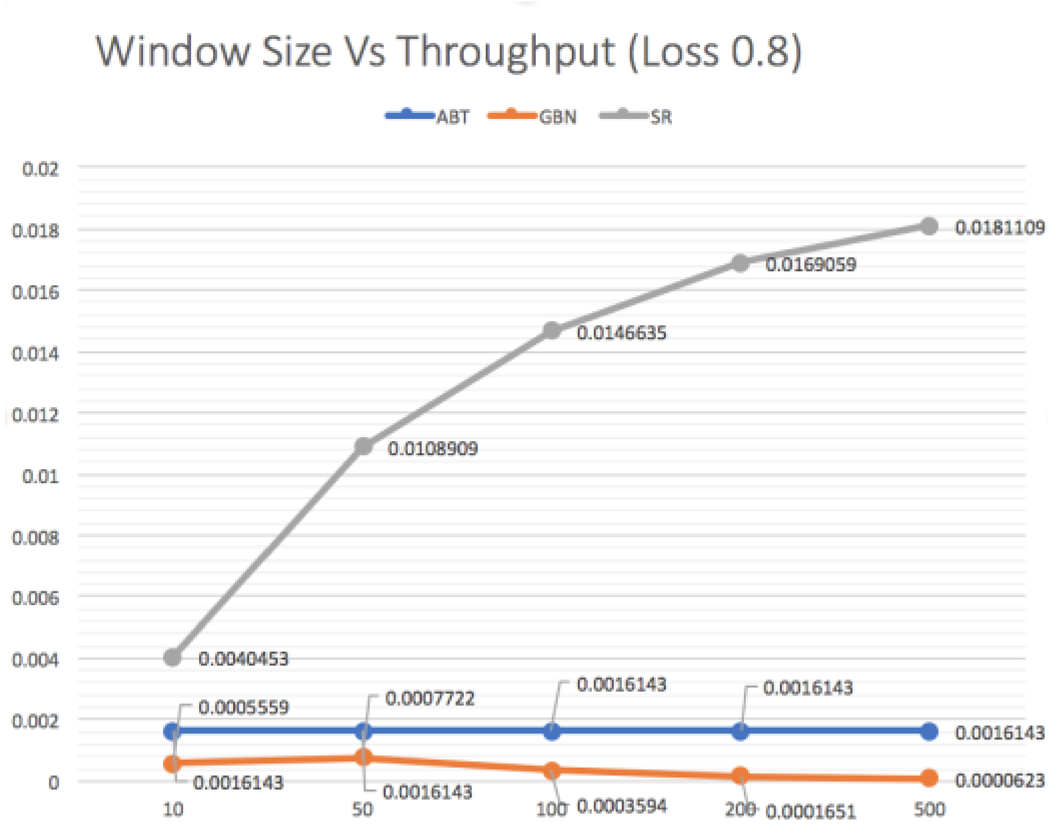
Expected Behavior/Variations

a. ABT is maintaining constant throughput as expected
b. SR has a better throughput and almost constant output for various window sizes as expected
c. GBN has a decreasing throughput upon increasing window size, this is because of retransmission of entire window when there is a loss

Conclusion

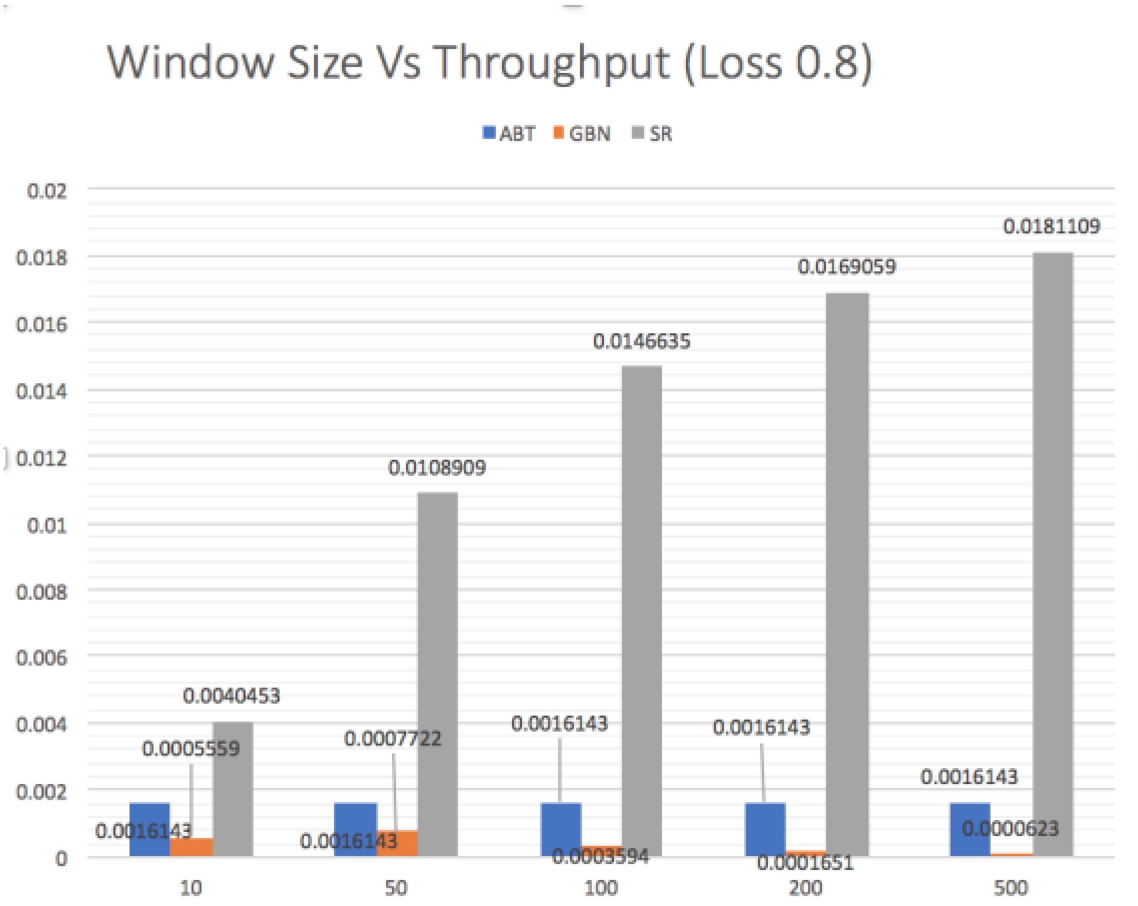
a. GBN is preferable when loss probability is very less and window size is small although SR has slightly better throughput than GBN. This is because of the implementation cost and buffers which are required in the receiver side
b. SR is preferable if bandwidth is of high priority

4.3 Graphs For Loss probability 0.8



Graph 9

X Axis : Window Size; Y Axis : Throughput Value



Graph 10

X Axis : Window Size; Y Axis : Throughput Value

General Observations (Loss 0.8)

a. ABT throughput remains constant for various window sizes as it does not depend upon window size
b. GBN throughput has a better throughput for window size 50 and then decreases
c. SR throughput is increasing as window size increases depicting a different behavior

Expected Behavior/Variations

a. ABT is maintaining constant throughput as expected
b. SR has a better throughput but the throughput is increasing when the window size increases showing an anomaly.
c. GBN has a decreasing throughput upon increasing window size, but increases just for a window size of 50 depicting an anomaly

Final Conclusion

a. GBN is preferable when loss probability is very less and window size is small although SR has slightly better throughput than GBN. This is because of the implementation cost and buffers which are required in the receiver side

b. SR is preferable if bandwidth is of high priority