Analysis Report

Avinash Palamanda **50166647**

I have read and understood the course academic integrity policy.

Timer value Used :

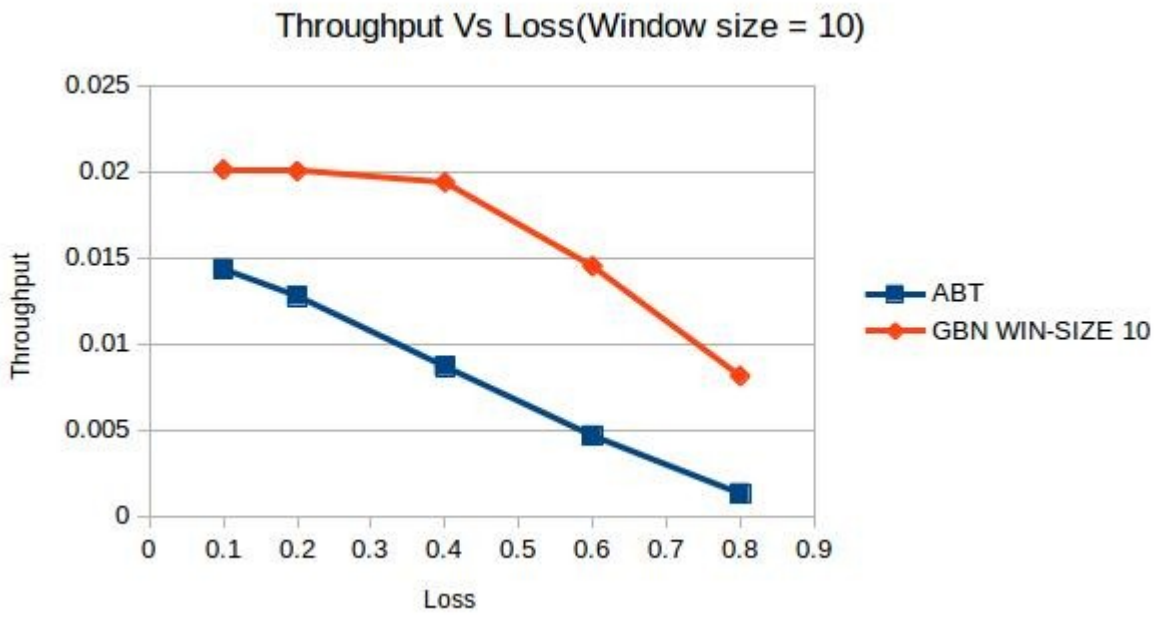
I have used a timer value of 20 units. The reason for using this timer value is that in case of large window size if the whole window needs to be transmitted again in such cases there may be lot of retransmission in the medium hence it is better to give large RTT value. In case if we provide smaller one then there may be more retransmitted packet .

In my code the throughput is good when the RTT is optimized for 20.0.

Experiment 1:

For Window Size 10:

LOSS	ABT	GBN WIN-SIZE 10
0.1	0.0143422	0.0201073
0.2	0.0127784	0.0200198
0.4	0.0086873	0.0193737
0.6	0.0046725	0.0145094
0.8	0.0013136	0.0081303



From the graph we have obtained we observe the following:

Alternating Bit: For alternating bit case we are seeing that as the value of loss increases the value of throughput decreases.

As the value of loss increases the message delivered from A gets lost. This causes timeout and the number of packets received in the B side also decreases. Since ABT is stop and wait it doesn't transmit new packet until ack is received for old one.

The other problem in the ABT is that though the A has delivered the packet to B.The ACK packets from B can also be lost. This causes the A to retransmit the packet.

Hence the output plot obtained stays true.

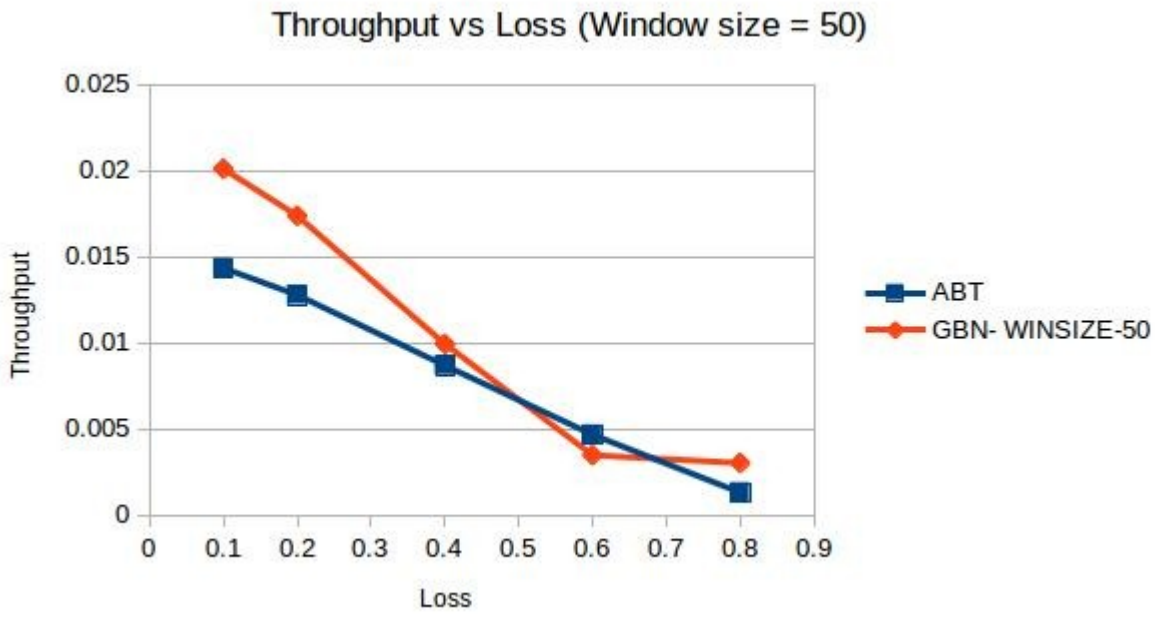
Go Back-N: In case of GBN ,as the value of loss increases the throughput decreases. But in this case the throughput is much higher than that of the Alternating bit.

The throughput is higher because in case of packet loss the GBN transmits all the packets in windows which has not been acknowledged. Hence there is more number of packets being transmitted from the transport layer rather than the ABT where only one packet is transmitted.

Experiment 2:

For Window Size 10:

LOSS	ABT	GBN- WINSIZE-50
0.1	0.0143422	0.0201008
0.2	0.0127784	0.017377
0.4	0.0086873	0.0099312
0.6	0.0046725	0.0034796
0.8	0.0013136	0.0030268



From the graph we can conclude the following:

Alternating Bit: This is same as the previous given explanation.

Go Back-N – 50:

We are seeing that GBN is better when there is low loss and corruption. This is the case because the GBN sends 50 packets at a go.

But as the loss increases we are seeing that GBN and ABT has about same value of throughput. The throughput decreases in GBN because the number of retransmission is more.

This case might be that B received the message and it has also sent the ACK which might have been lost or corrupted. But A would resend that packet.

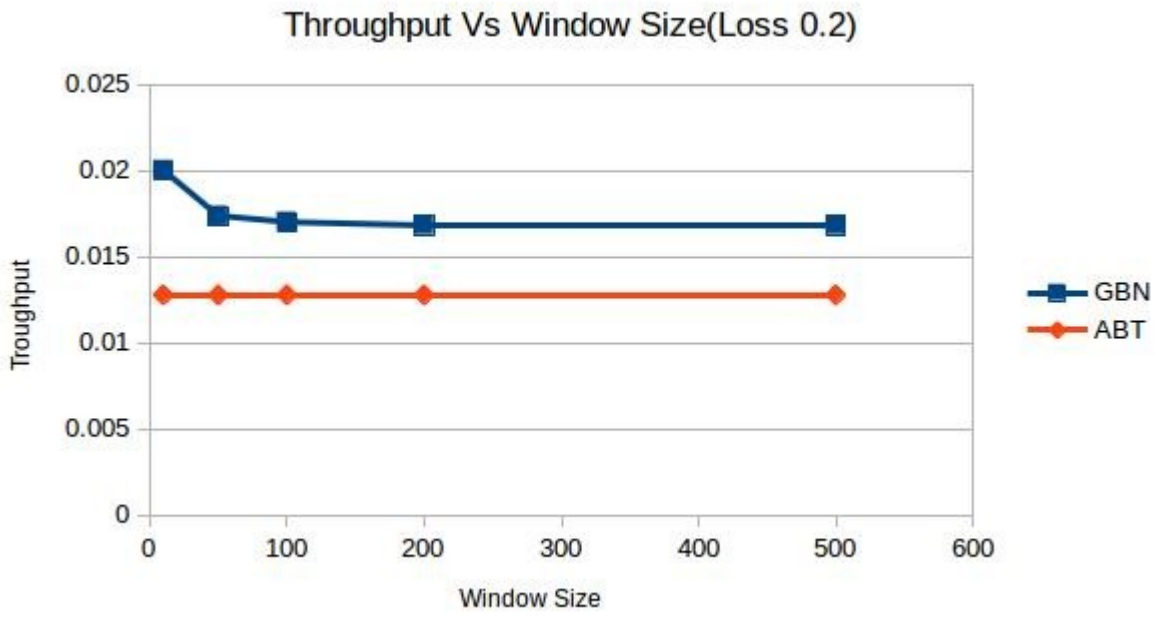
That is if the first packet in the window fails to receive ACK and also the remainder packets also doesnt get ACK,in such case the whole window is again transferred.

Hence there is a drastic decrease in the throughput.

Experiment 2:

Loss – 0.2 :

Window Size	GBN	ABT
10	0.0200198	0.0127784
50	0.017377	0.0127784
100	0.0170192	0.0127784
200	0.016832	0.0127784
500	0.015482	0.0127784



Alternating Bit:

In case ABT it doesn't depend on Window size. Hence we have static throughput in graph.

Go Back-N:

We are seeing when the window size is lower, the throughput value is higher. For an example when N=10 we have higher throughput than ABT case (0.02) but as the value of N increases the throughput deceases.

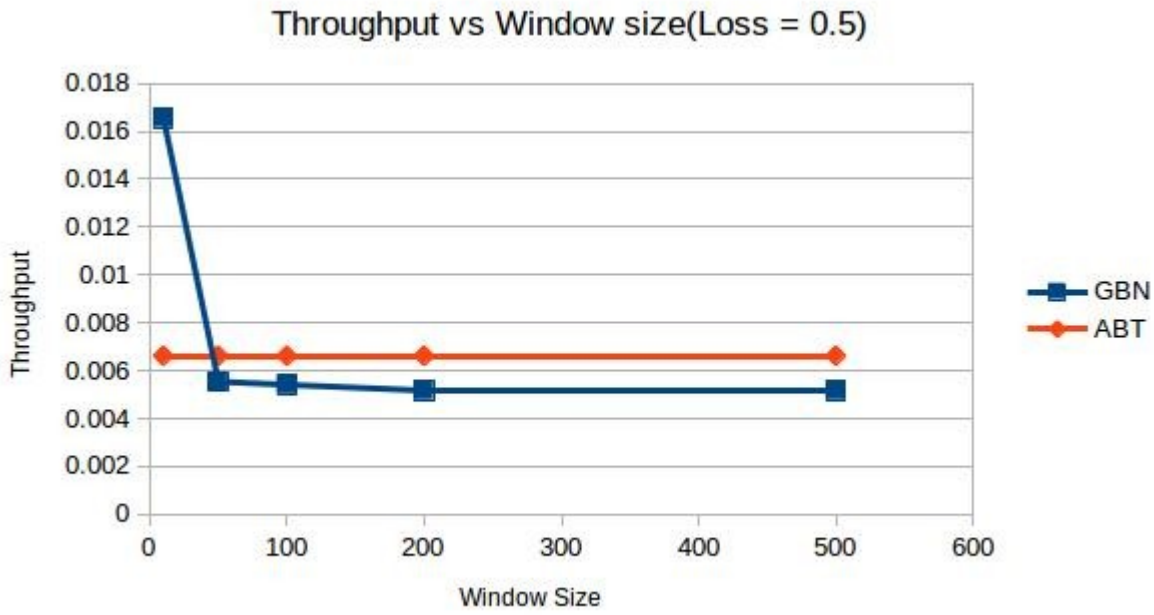
The reason behind this is as N increases the number of retransmission of packet increases for each lost packets,hence the transport layer retransmits more number of packets.

But these retransmitted packets might have been Acknowledged by the B and ACK might have been lost. Thus the B just drops those retransmitted packet and send ACK message again.

Thus loss of those ACK and Retransmission of Message Packets causes decrease in throughput.

Loss – 0.5:

Window Size	GBN	ABT
10	0.0165599	0.0066147
50	0.0055352	0.0066147
100	0.0054093	0.0066147
200	0.0051626	0.0066147
500	0.0051642	0.0066147



Alternating Bit:

We are seeing that throughput remains same but lower than previous case since there is increase in loss.

Go Back-N:

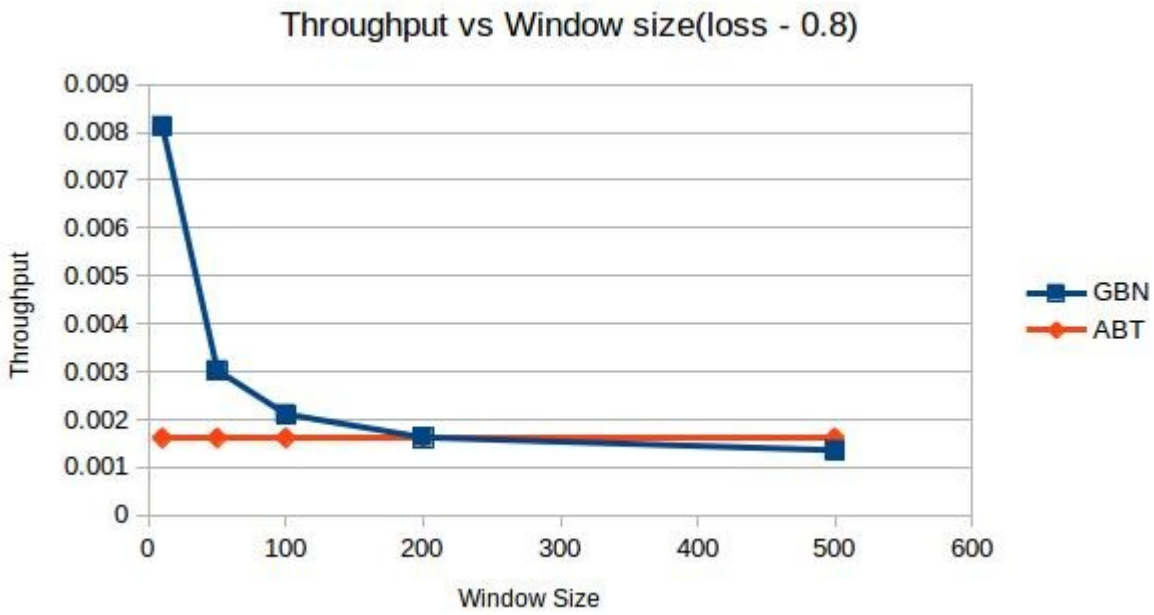
We see same kind of situation as previous but in there is one difference ,we see that as the value of window increases the throughput goes below ABT(more or less same value).

The reason behind this in case of 0.5 probability nearly 50% of packets are lost. Hence there is lot of packets from application moves into buffer. There will also be lot of retransmission. These retransmitted packets might have already reached B and B would have also sent ACK packet these ACK might have been lost. The retransmission also causes message from buffer not be sent to B.

All these factors will result in decrease in throughput as all packets from Application layer from A would not have reached Application layer of B

Loss – 0.8:

Window Size	GBN	ABT
10	0.0081303	0.0016136
50	0.0030268	0.0016136
100	0.0021107	0.0016136
200	0.0016241	0.0016136
500	0.0015802	0.0016136



Alternating Bit:

We are seeing that for ABT the throughput value is constant and is very low. This is because ABT is stop and wait protocol so it waits for ACK for each message. When loss is 0.8 and Corruption 0.2 more than 80% of packets are lost or not correct and hence there will lot of retransmission of same packets and other packets from Application layer will be dropped directy.

Go-Back -N:

We are seeing that for lower window size the throughput is much better than that of the large window size.

Since loss is 0.8 about 80% of the packets will be retransmitted from the A's end and in same way at B's end those packet might have reached but ACK from B would have been lost. Hence some of the retransmitted packet would be directly dropped from the B's end.

Hence when the window size increases there might be lot of retransmission. For example when window size is 500 there is a possibility of more than 400 messages being retransmitted at time.

This behavior of GBN causes decreases in throughput with increase in window size for lossy medium.

Hence we can conclude that for heavy lossy medium it is better to have GBN with low window size.