CSE 589 Modern Networking Concepts (FALL '20)

Aditi Pratap Ghatge - 50338244
Kamalesh Ram Chandran Govindaraj - 50338246

Acknowledgement

I (We) have read and understood the course academic integrity policy located under this link: https://cse.buffalo.edu/~lusu/cse4589/

Timeout scheme

All experiments listed below are using a timeout scheme of 50 units. Taking timeout values too small will cause unnecessary packet retransmission. The reason for packet delay can be anything due congestion or queuing. Making it too long will avoid packet retransmission but if some packet is actually lost we don't want to wait for too long.

Timer implementation

ABT: For implementing this protocol, a single timer is required and it is started at the time of transmission of each packet. The timer is stopped only when an acknowledgement of the packet is correctly received by the sender. Further, if there any outstanding packets in the buffer, the oldest of them is transmitted by the sender and the timer is started again. If the timer expires or upon receiving a corrupt acknowledgement, the current packet is retransmitted and the timer is restarted.

GBN: Go-Back-N protocol requires one single timer for a whole of the window so it isn't as complicated as SR protocol. We start the timer when we send the base packet of the window. Only when the base packet including all other packets are received the timer is stopped. If just the base packet is received in the window, the base is incremented and the window is shifted by one and the timer is started again to send the packets of the window size from the buffer.

SR: Selective Repeat protocol requires multiple timers to keep track of each of the packets sent within the window but as we're provided with only one physical timer, we have got creative in using the same physical timer to keep track of all packets in a window. We created timestamps for each of the packets within the window and waited for the acknowledgement. When we received acknowledgement for the base packet (base packet is the first packet sent in that particular window), we increment the base, stop the physical timer and calculate a new timeout value to start the timer for the rest of the unacknowledged packets. We calculate the new timeout by subtracting the timestamp of the base packet from the timeout value we have set for the physical timer. This new timeout will be used to start the timer again until all other packets have been received. This same concept of timer is used in timer interrupt too.

Comparing Performance of the Three Protocols

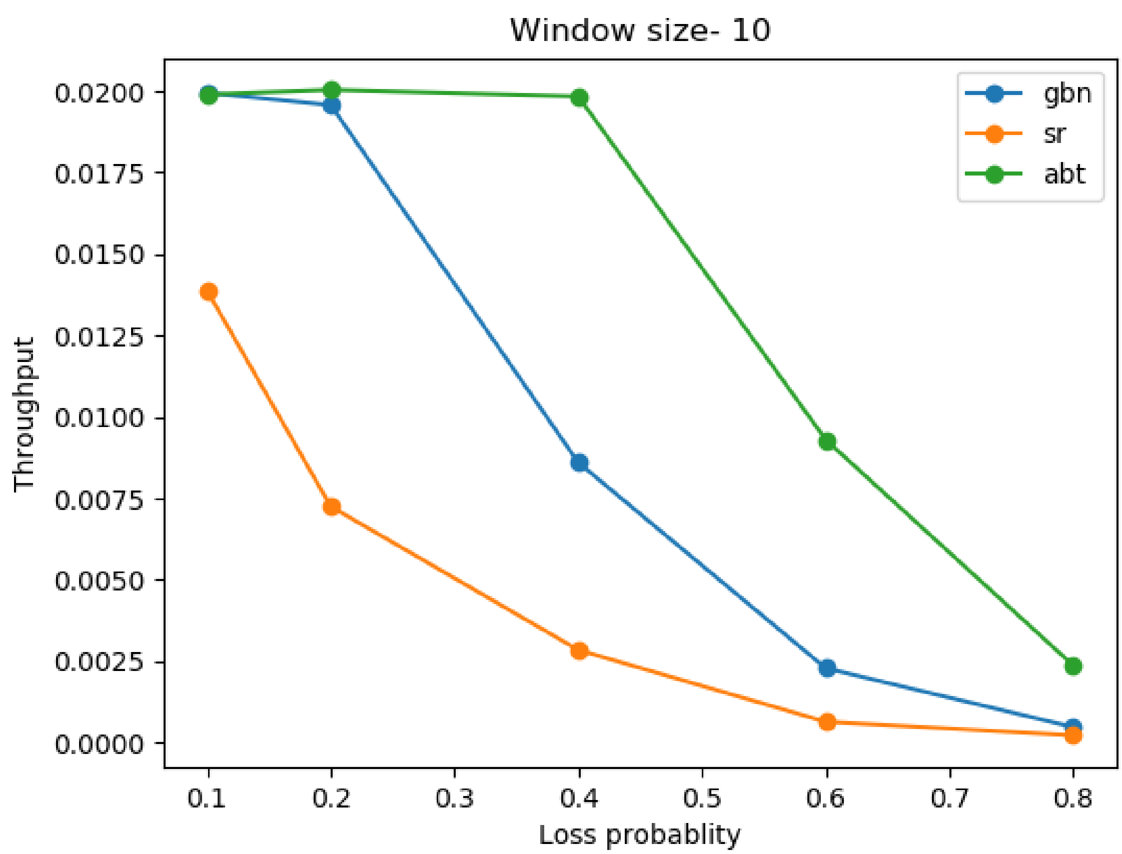
The following simulator parameters are in common for all experiments:

i. Number of messages sent by Host A: 1000
ii. Mean time between messages from Host A's application layer: 50
iii. Probability of packet corruption: 0.2

Experiment 1: Comparing the throughput for the three protocols with packet loss probabilities {0.1, 0.2, 0.4, 0.6, 0.8} and window sizes (for GBN and SR) {10, 50}

1. Window Size = 10

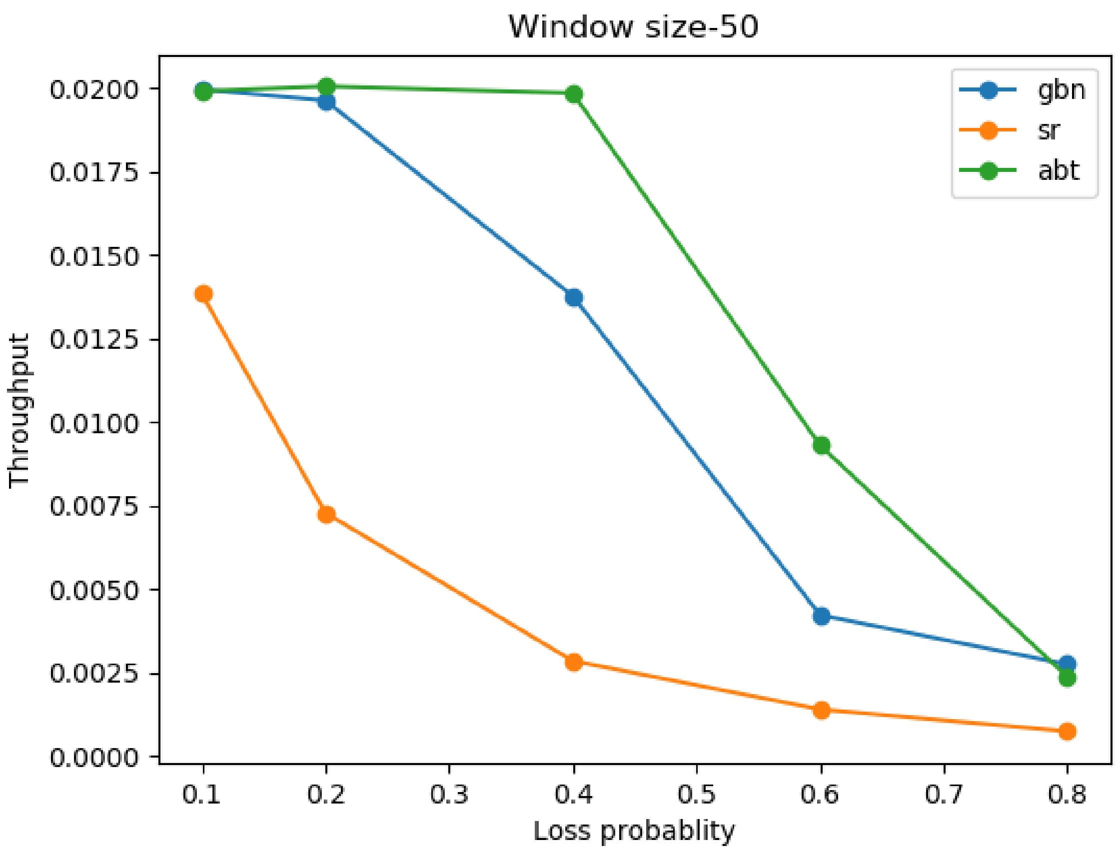
	0.1	**0.2**	**0.4**	**0.6**	**0.8**
GBN	0.0199439	0.019571	0.0086	0.002287	0.000481
ABT	0.0198961	0.020036	0.019836	0.0093	0.002371
SR	0.013882	0.007252	0.002848	0.000644	0.000228



ABT is performing better here as compared to other protocols. Real expectation was that GBN and SR perform similarly. For every algorithm performance is decreasing with increase in the loss value. Even with increase in loss value up to certain point ABT's performance was stable. In the case of GBN the performance decreased drastically with the increase in the loss value. In case of SR one thing can be noticed is after a certain value of loss here its 0.6 increase in loss does not matter after a certain point because throughput remained constant after that but initially throughput of SR was more sensitive to loss than any other protocol.

2. Window Size = 50

	0.1	0.2	0.4	0.6	0.8
GBN	0.0199439	0.019613	0.013766	0.00422	0.002746
ABT	0.0198961	0.020036	0.019836	0.0093	0.002371
SR	0.0138298	0.007279	0.002849	0.001399	0.00074

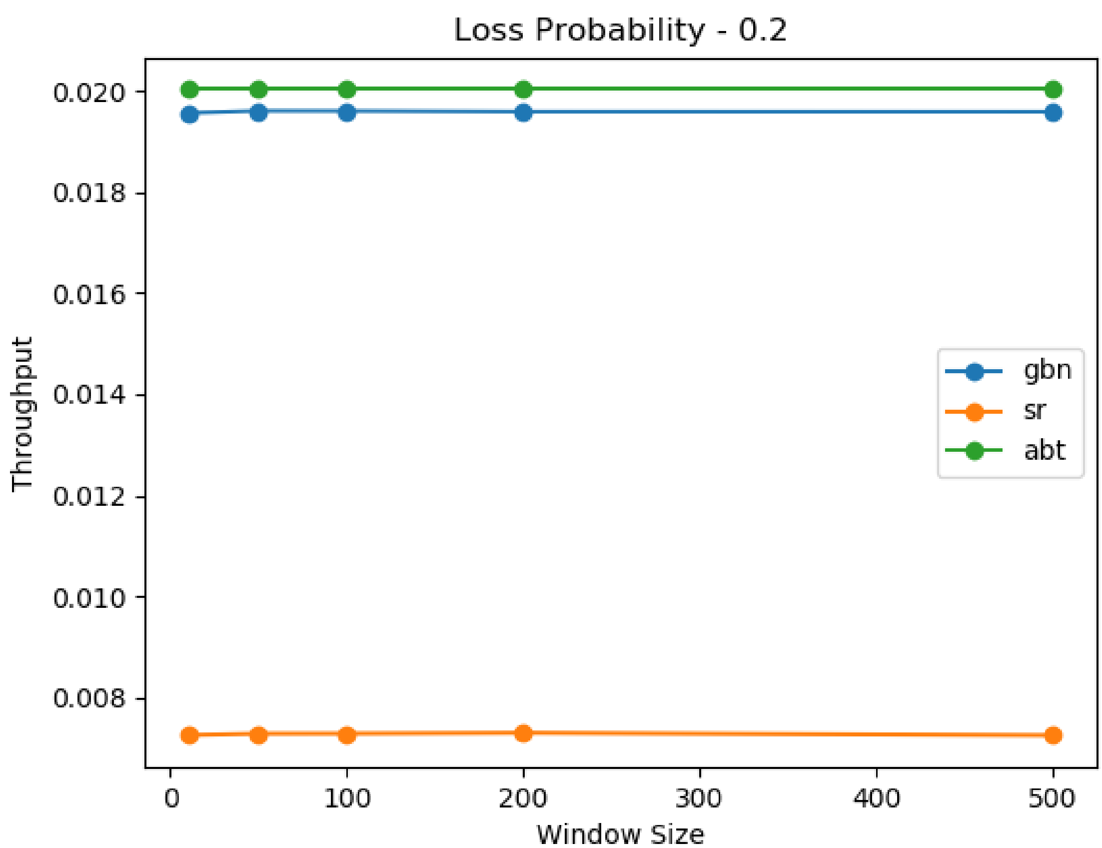


As window size increased from 10 to 50 there is no significant increase in throughput visible for any protocol. GBN performed a little better than it performed with window size 10 as we can see with increase in window size there is less drastic decrease in performance. Behavior of SR is almost similar to what was with window size 10. In case of GBN timeout value can affect the performance because up to certain point if we increase the window size the throughput increases. I conclusion increase in window size affected GBN little bit but did not affect SR and ABT.

Experiment 2: Comparing the throughput for the three protocols with packet loss probabilities {0.2, 0.5, 0.8} and window sizes (for GBN and SR) {10, 50, 100, 200, 500}

1. **Packet Loss Probability = 0.2**

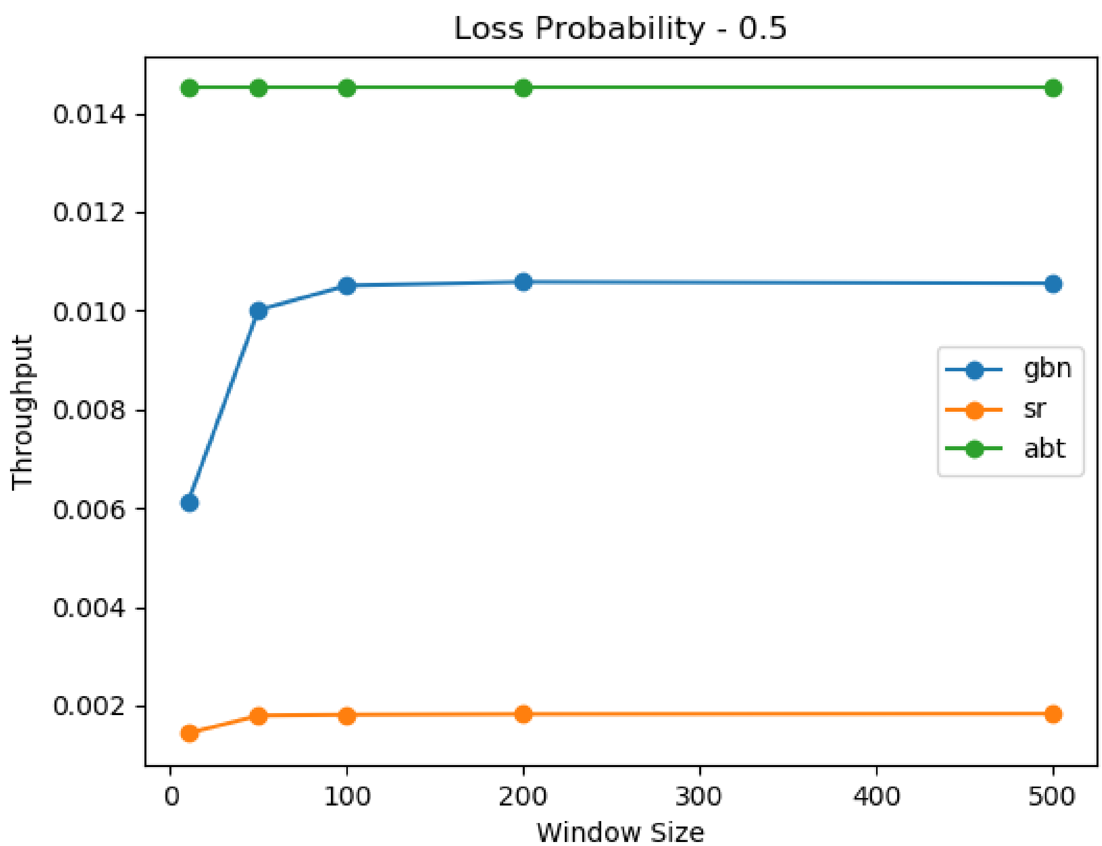
	500	**200**	**100**	**50**	**10**
GBN	0.0195963	0.019598	0.019611	0.019613	0.019571
ABT	0.0200359	0.020036	0.020036	0.020036	0.020036
SR	0.007248	0.007294	0.007281	0.007279	0.007252



Selective repeat performs similarly at all the window sizes this is because the packets are not coming as fast for SR in order to use the window size fully. Whereas GBN throughput is expected to increase with the window size but with loss 0.2 it is not showing any performance increase but it's stable.

2. Packet Loss Probability = 0.5

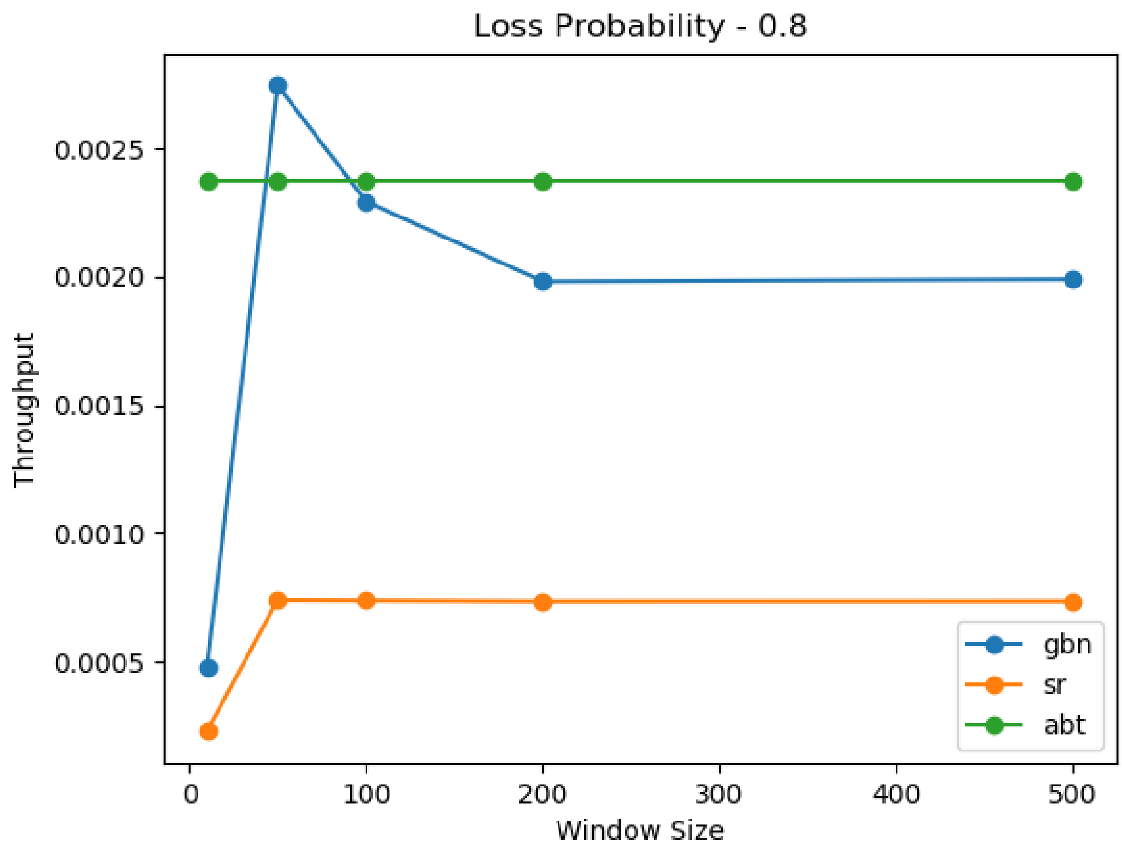
	500	**200**	**100**	**50**	**10**
GBN	0.0105552	0.010582	0.010506	0.010014	0.006106
ABT	0.0145122	0.014512	0.014512	0.014512	0.014512
SR	0.0018357	0.001827	0.001813	0.001796	0.001439



ABT does not show any change in performance with the increase in the performance with the increase in the window size. But GBN showed drastic change in performance when the window size changed from 10 to 50 then gradual increase up to window size 110 after that performance is stable even after increasing the window size, so after a certain point for GBN as well change in window size did affect the performance that much. In case of SR the change in window size took little bit effect on the performance while window size changed from 10 to 50 but after that performance was stable irrespective of the window size. As compared to throughput with loss 0.2 the performance is declined in every other protocol.

3. Packet Loss Probability = 0.8

	500	200	100	50	10
GBN	0.0019904	0.001982	0.002293	0.002746	0.000481
ABT	0.0023707	0.002371	0.002371	0.002371	0.002371
SR	0.000736	0.000736	0.000739	0.00074	0.000228



When loss is 0.8 we can see there is a drastic increase in the performance in case of GBN and SR after a certain window size. But in case of GBN it started performing poorly when loss rate and window size both are large. The main reason for this is retransmission of lost packets again and again. In case of ABT window size does not matter hence the performance is stable even after increasing the window size but we can clearly see due to increase in loss probability the throughput is dropped compared to loss probability of 0.5. In SR if the window size is increased the throughput is constant after window size became 50 that is because packets are not arriving as fast for SR to use the window size fully.