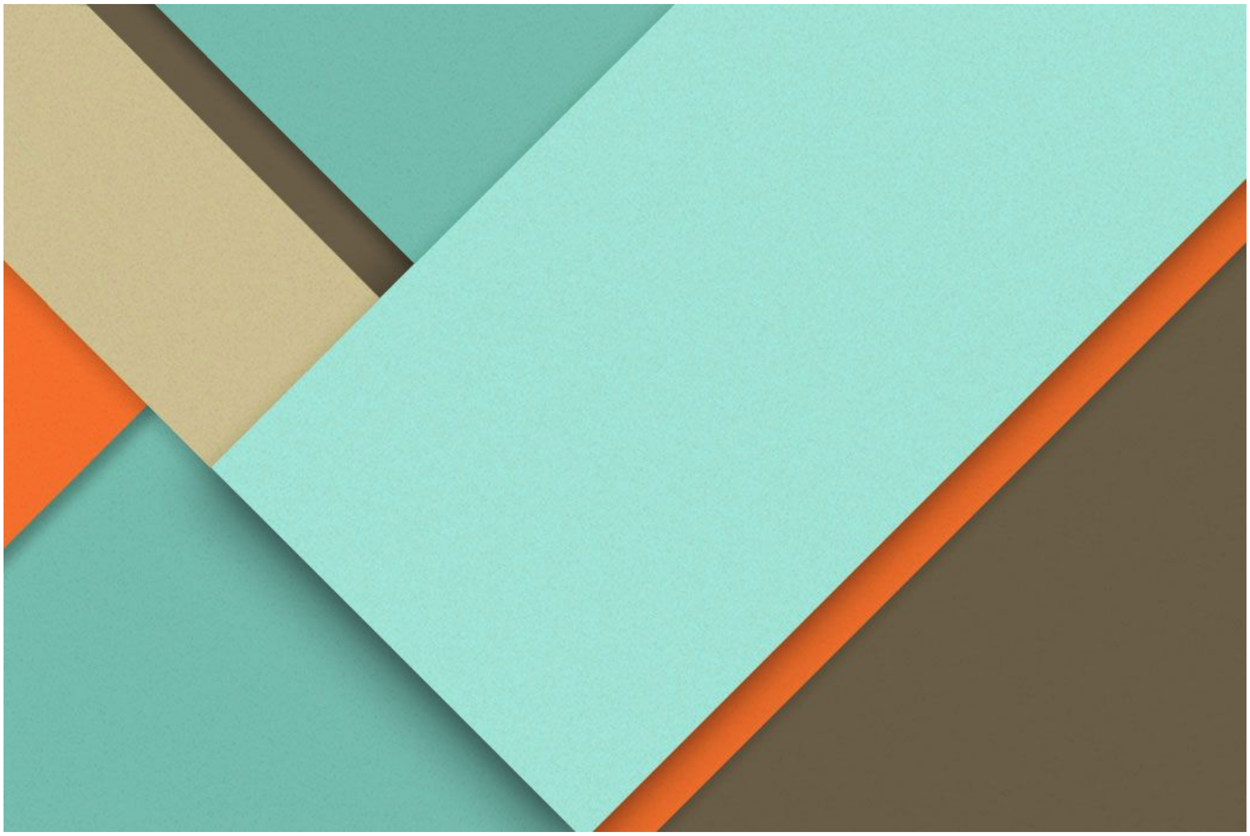


Reliable Transport Protocols Analysis

11/03/2017

I have read and understood the course academic integrity policy.

Timeout Scheme:

1. Alternating-Bit-Protocol (ABT) : A constant timeout value of 18.0 time units is used for this protocol. I ran experiments with multiple time-out values for different loss rates and observed that this timeout value gave maximum throughput.
2. Go-Back-N (GBN) : A constant timeout value of 20.0 time units is used for this protocol. I ran experiments with multiple timeout values for different loss rates and window size and observed that this timeout value gave maximum throughput for this protocol. I was planning to implement different timeout values for each window size but I observed that this timeout value suited for all the window size.
3. Selective-Repeat (SR) : A constant timeout value of 18.0 time units is used for this protocol. Similar to GBN, I observed that this timeout value was suitable for different loss rates and window size.

Multiple Software Timer in SR :

Every packet that is sent out is assigned a TIMEOUT_VAL i.e 18.0 time units. A timer with timeout value of 1 time unit is started if there were no in transit packet. Every time timeout happens, the TIMEOUT_VAL assigned to all the packets in transit is decremented by one time unit and checked if there are any in transit packets that have timed out. If there is any packet that has timed out, the packet is retransmitted and again TIMEOUT_VAL is assigned to that packet. Finally, if there are any in transit packets, timer with timeout of one time unit is started.

A list of structure (struct node), whose first element is pointed by winHead_A, is used to buffer the packets at sender side.

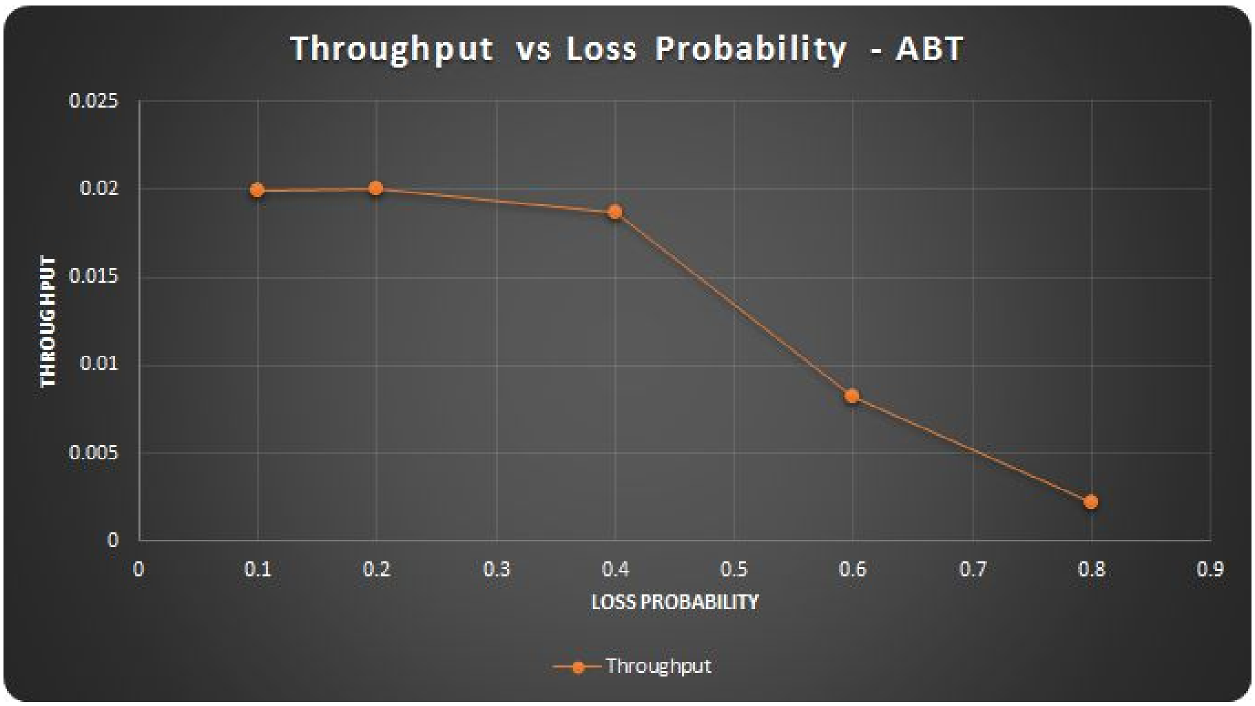
```
struct node {
        struct node* next;       //Points to next element in the list.
        struct pkt pktBuf;       //Packet to buffer.
        int timeOut;             //Timeout value of the packet.
        bool bAckReceived;       //Flag to denote if ACK is received for this packet.
}
```

TIMEOUT_VAL is assigned to timeOut after sending the packet to layer 3. timeOut is decremented for every timer interrupt that is triggered every time unit. When timeOut value reaches 0, the packet is retransmitted and again TIMEOUT_VAL is assigned to timeOut.

Experiment 1:

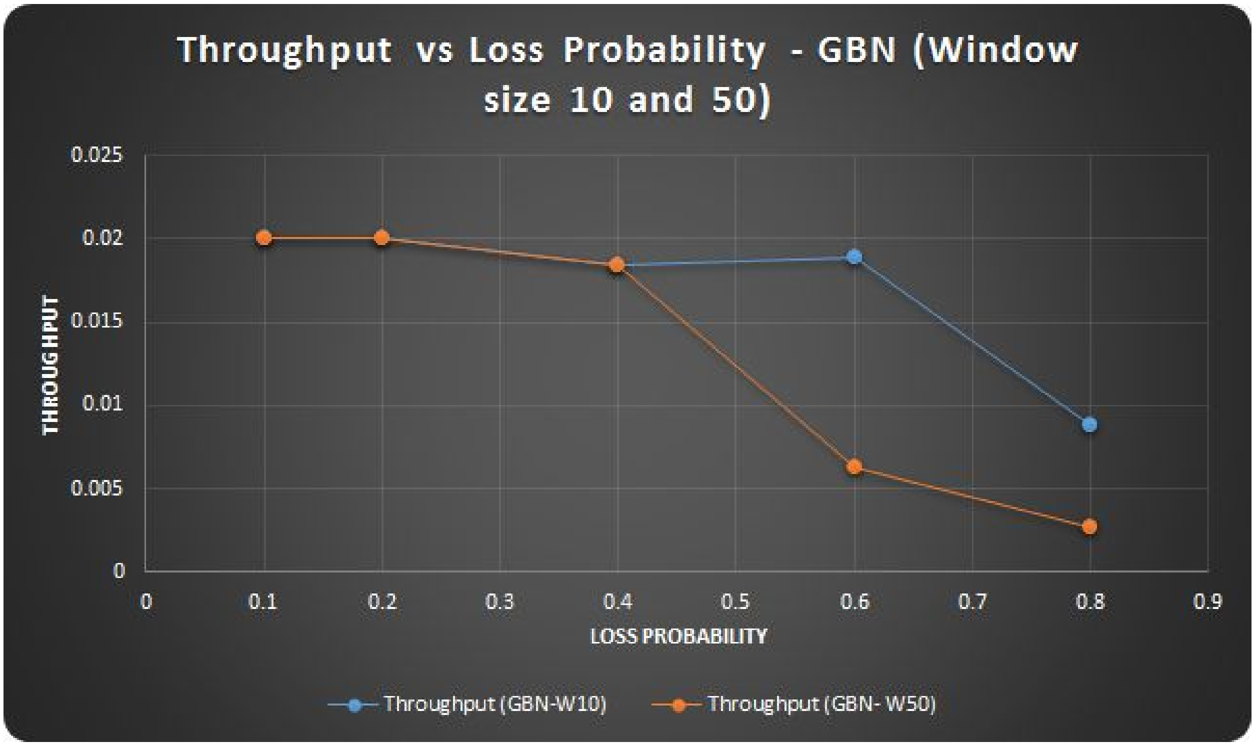
The below experiments are executed with a mean time of 50 between message arrivals (from A's layer 5) and a corruption probability of 0.2.

I. Alternating-Bit-Protocol (ABT):



Analysis: This experiment was run for ABT with loss probabilities of 0.1, 0.2, 0.4, 0.6 and 0.8. We can see from the above graph that as the loss probability increases, throughput decreases i.e. throughput is inversely proportional to loss probability. A new packet is not sent until sender receives ACK for the previous packet. When the loss probability is high, a single packet is lost multiple times and in addition, ACK of that packet might also be lost. Due to which, the time taken to successfully send a packet is high and the next packet has to wait for a longer time. Hence, the throughput decreases as the loss probability increases.

II. Go-Back-N (GBN):

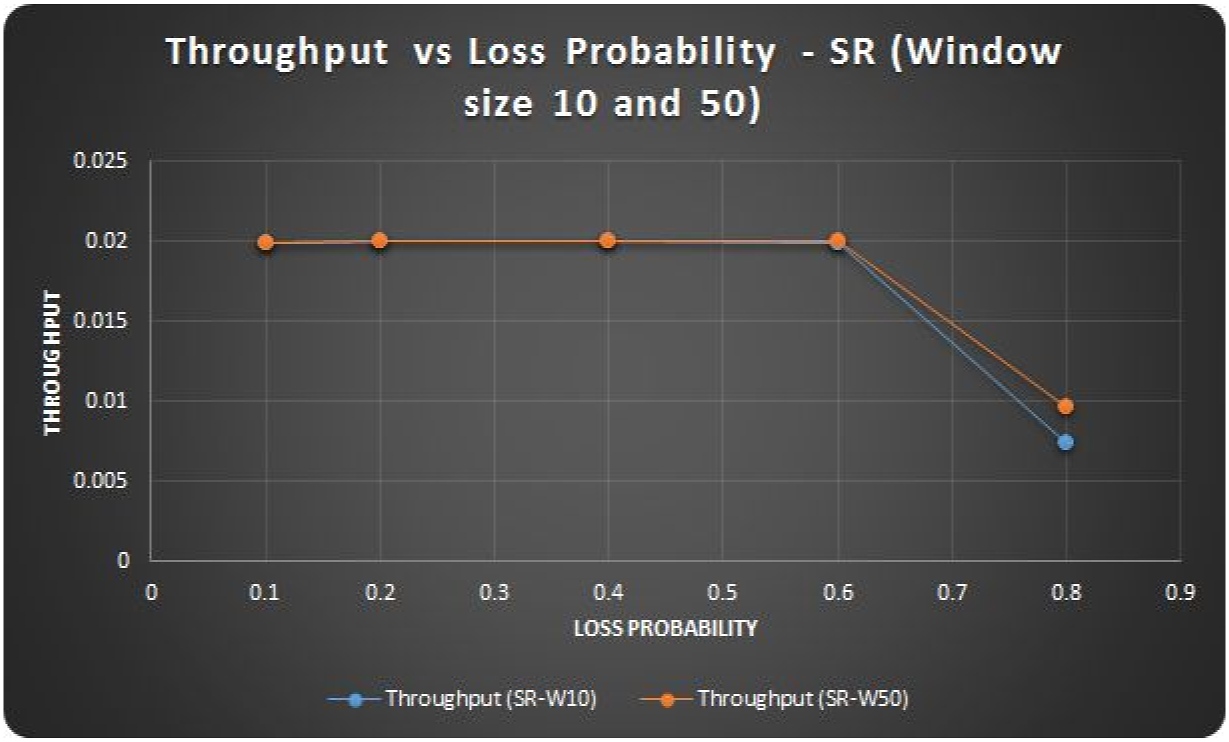


Analysis: This experiment was run for GBN with loss probabilities of 0.1, 0.2, 0.4, 0.6 and 0.8 and window size of 10 and 50. The receiver side for GBN does not buffer the messages received out of order. Consequently, if any packet is lost, the sender has to retransmit all the packets that were sent after the lost packet.

From the above graph, we can see that throughput decreases with increase in loss probability. In addition, when the window size is increased, throughput decreases drastically for higher loss probability. When the window size is high and a packet is lost, number of packets that need to be retransmitted is higher. Due to this, GBN gives lower throughput for higher window size with higher loss probability.

The throughput can be improved if the receiver buffers packets received out of order and when a packet is lost, the sender does not need to retransmit all the packets that were sent after the lost packet.

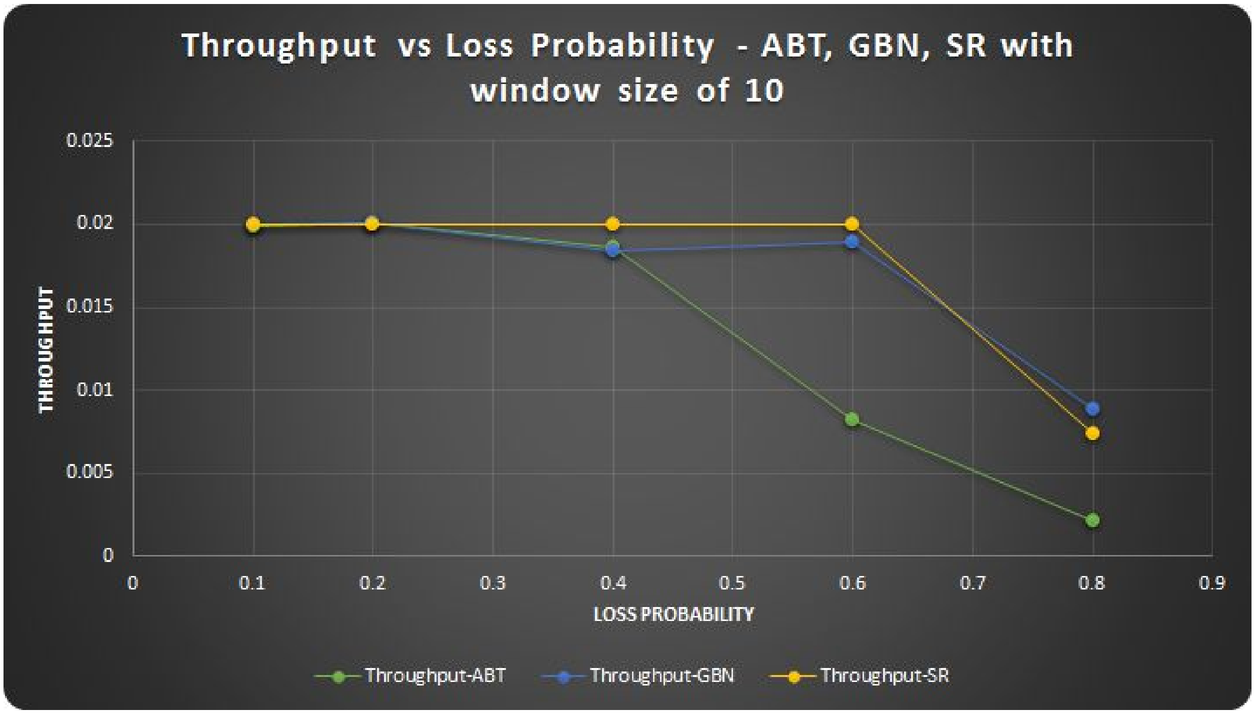
III. Selective-Repeat (SR):



Analysis: This experiment was run for SR with loss probabilities of 0.1, 0.2, 0.4, 0.6 and 0.8 and window size of 10 and 50. The receiver side for SR buffers the messages received out of order. The receiver sends ACK for packets received.

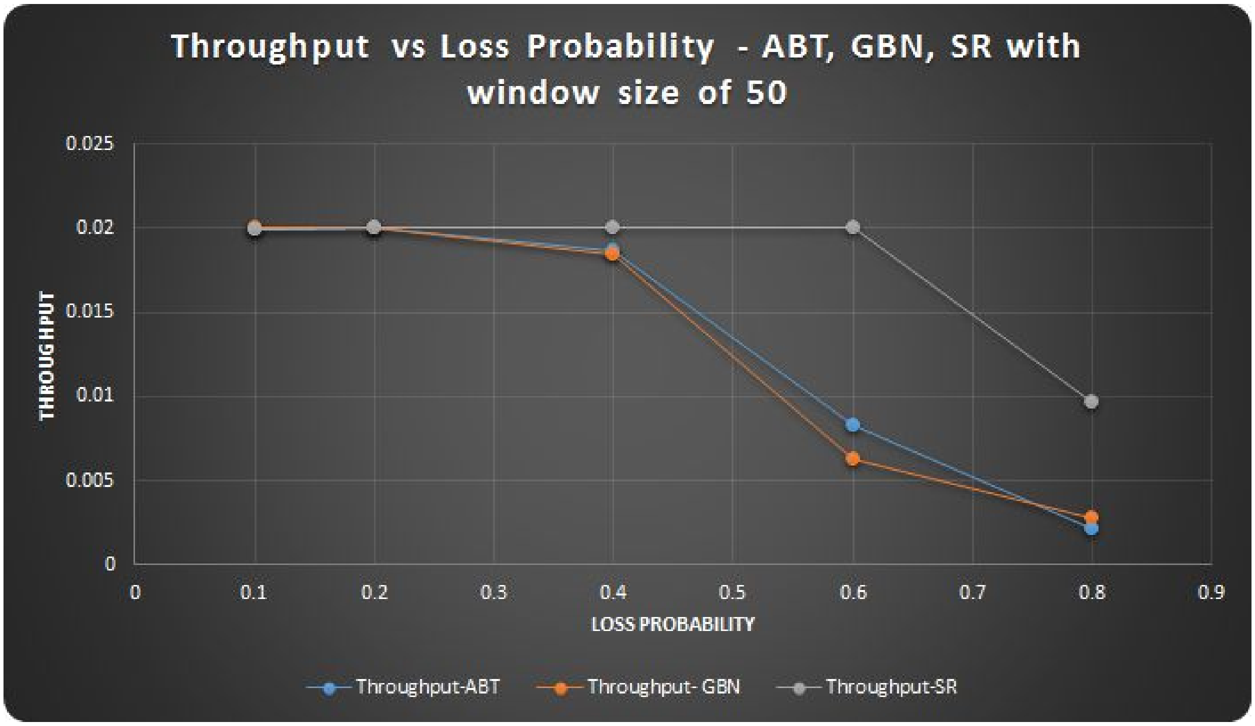
From the above graph, we can see that throughput decreases with increase in loss probability and the window size does not have significant impact on throughput. The lost packet is only retransmitted by the sender. As a result, throughput is not significantly affected by window size.

IV. Comparison of ABT, GBN and SR for window size of 10:



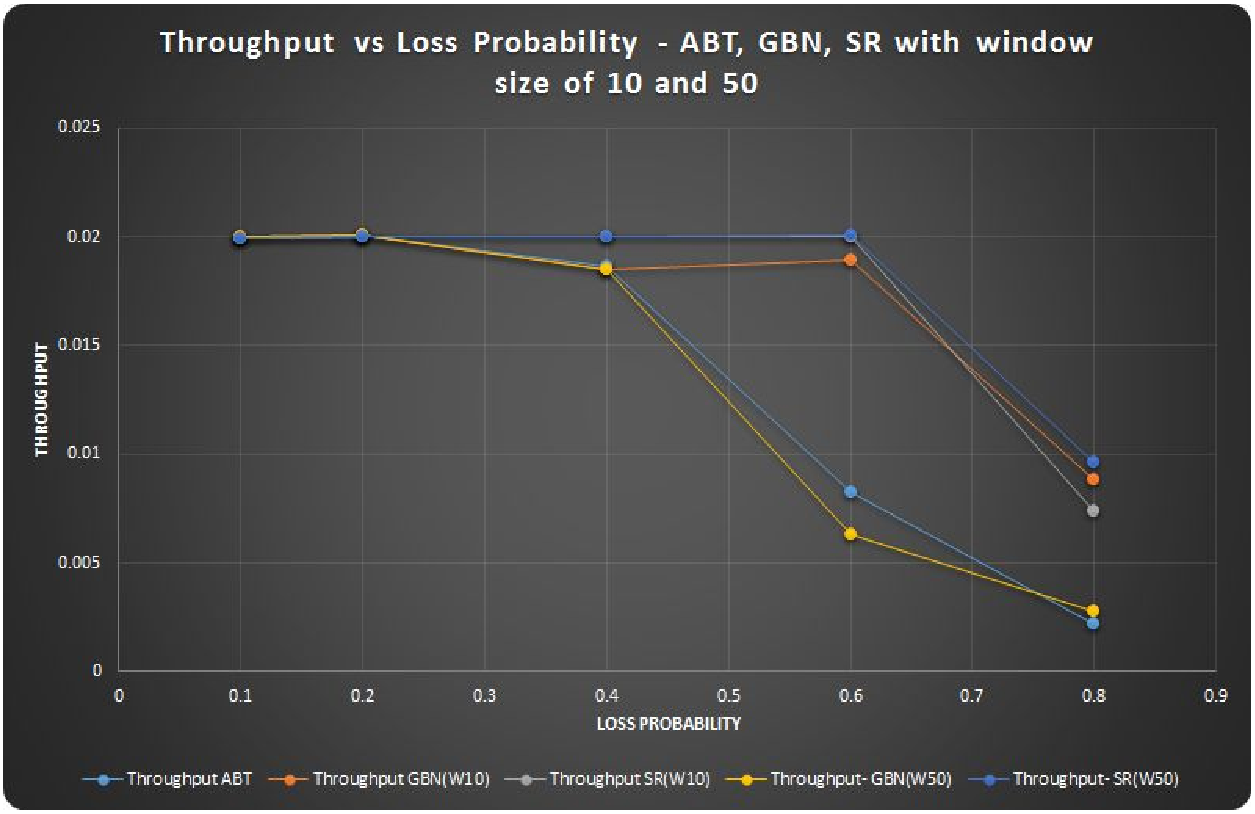
Analysis: From the above graph, we can clearly see that SR performs better that the other two protocols. The throughput for ABT decreases drastically as loss probability increases. The throughput of SR and GBN is comparable for window size of 10.

V. Comparison of ABT, GBN and SR for window size of 50:



Analysis: From the above graph, we can see that, SR performs much better than GBN and ABT for window size of 50. The performance of GBN decreases as window size is increased as seen from II. GBN needs to retransmit more packets if there is loss of packet and window size is high.

VI. Conclusion of Experiment 1:

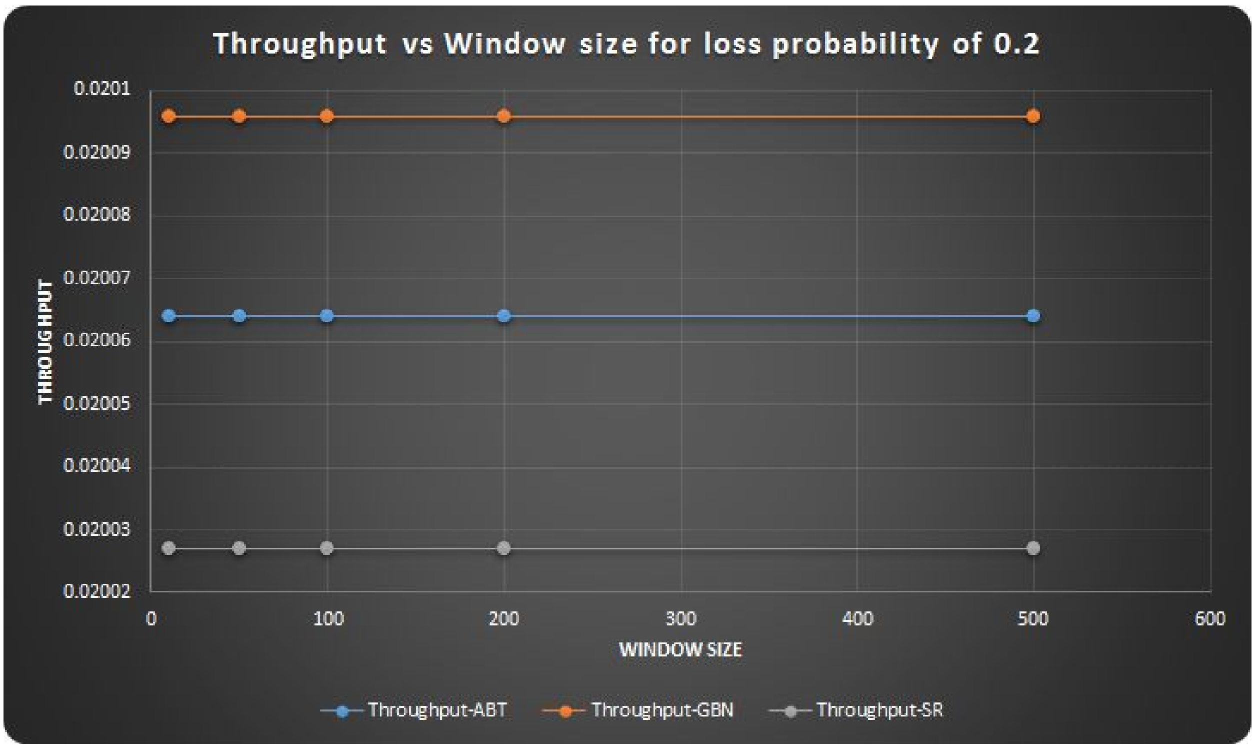


Analysis: We observed that throughput decreases with increase in loss probability for all three protocols. In addition, we observed that window size does not have much effect on SR but GBN is heavily impacted by window size as GBN has to retransmit more packet for a loss with increase in window size. ABT does not have a sliding window. Finally, we can conclude that the throughput of SR is better than other two protocols.

Experiment 2:

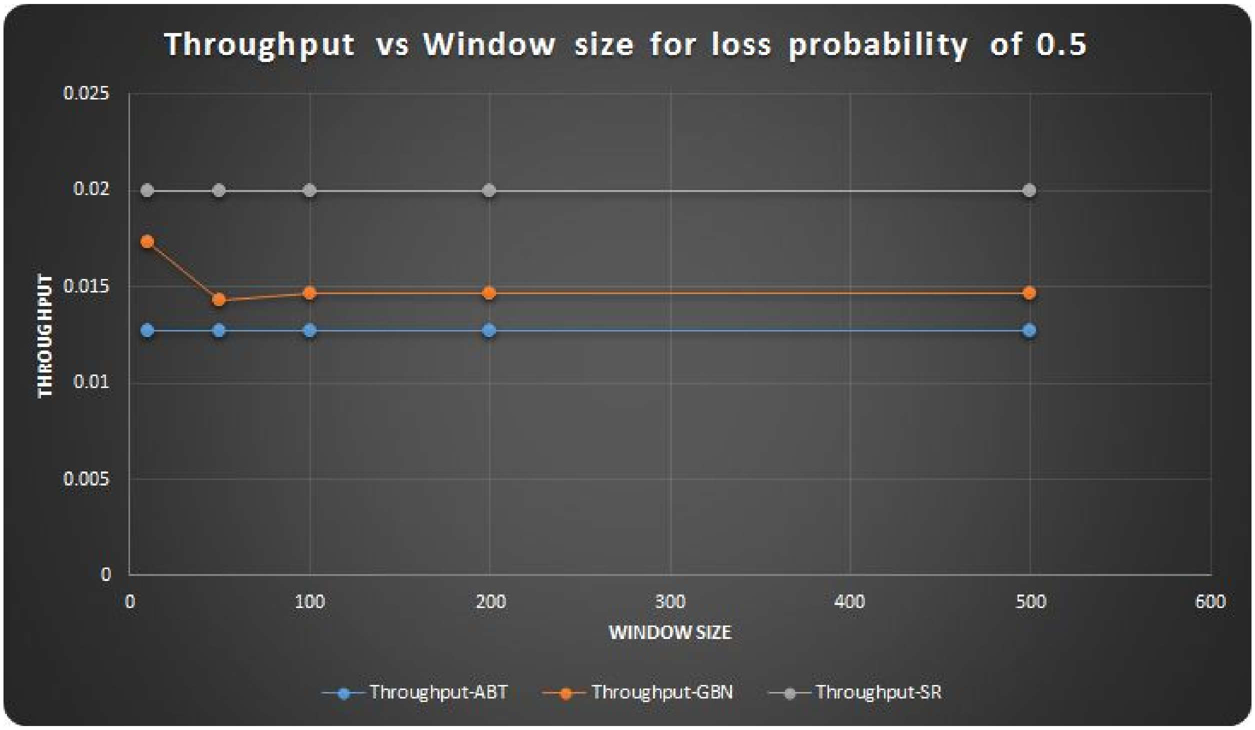
The below experiments are executed with a mean time of 50 between message arrivals (from A’s layer 5) and a corruption probability of 0.2.

I. Comparison of ABT, GBN and SR for loss probability of 0.2:



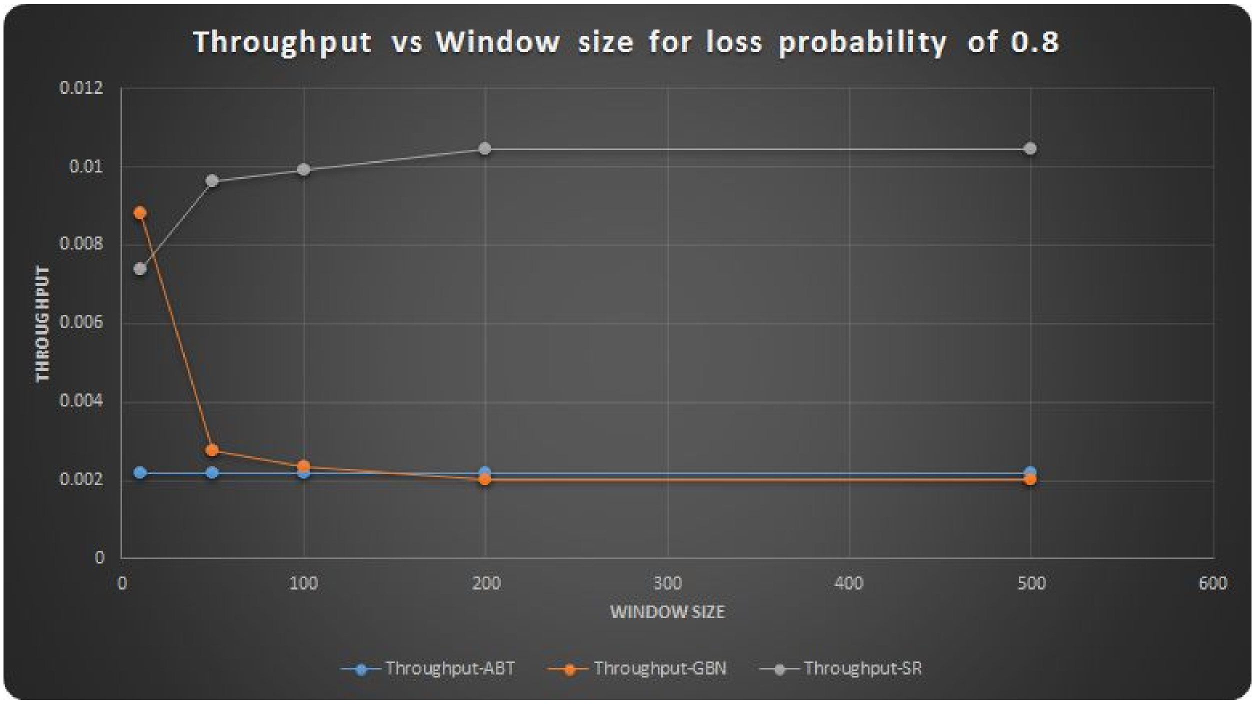
Analysis: This experiment was run for ABT, GBN and SR with loss probability 0.2 and window size of 10, 50 100, 200 and 500. ABT is not a sliding window protocol, so ABT does not have any effect for change in window size. From the above graph, we see that there is no change in throughput for different values of window size when the loss probability is low. Since there is low loss, GBN does not need to retransmit too many packets and window size does not make any difference. In case of SR, the window moves comparatively smoother as there are no loss and throughput remains constant regardless of window size.

II. Comparison of ABT, GBN and SR for loss probability of 0.5:



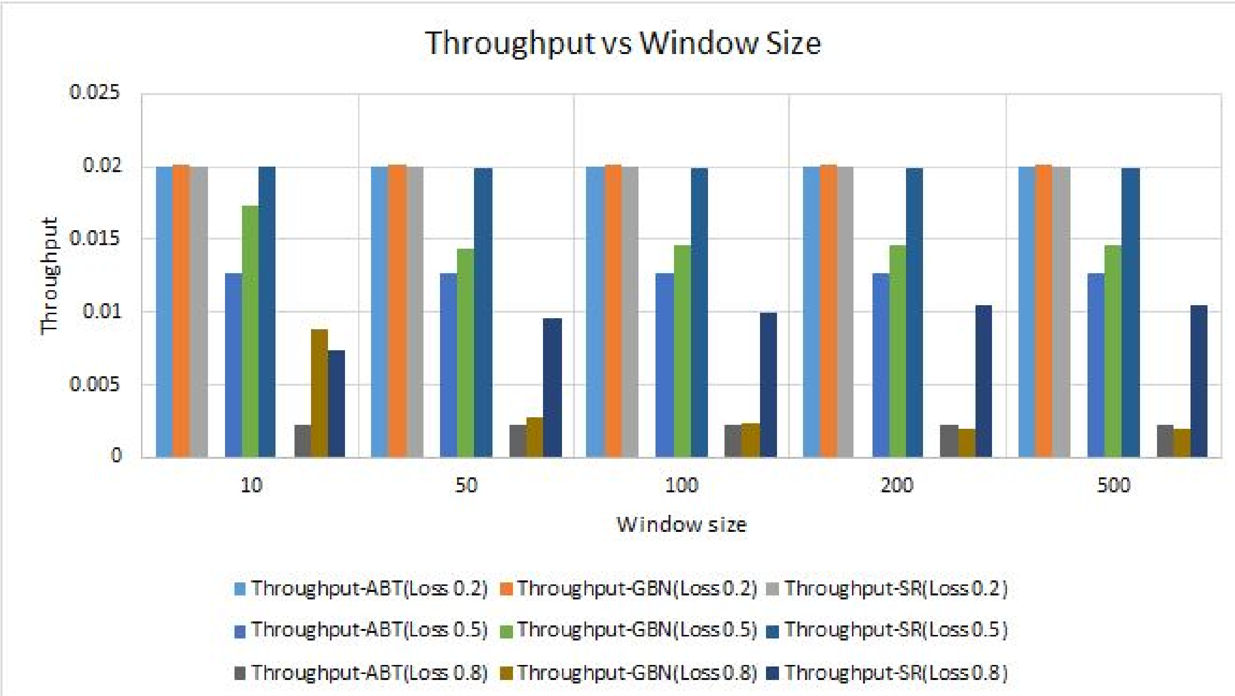
Analysis: This experiment was run for ABT, GBN and SR with loss probability 0.5 and window size of 10, 50 100, 200 and 500. Since ABT is not a sliding window protocol, ABT does not have any effect for change in window size. In case of GBN, the throughput decreased with increase in window size for a loss probability of 0.5. This is because more packets are lost and after timeout, whole window needs to be retransmitted. The number of packets that need to be retransmitted increase with the window size. So, the throughput decreased with window size and remained constant after window size of 100. Loss probability of 0.5 does not have any effect on throughput of SR for different values of window size. The sender retransmits only lost packets in case of SR. So loss of 0.5 does not affect SR for varying window sizes.

III. Comparison of ABT, GBN and SR for loss probability of 0.8:



Analysis: This experiment was run for ABT, GBN and SR with loss probability 0.8 and window size of 10, 50 100, 200 and 500. Since ABT is not a sliding window protocol, ABT does not have any effect for change in window size. In case of GBN, since the loss is high, the number of packets that need to be retransmitted increases with the increase in window size. So, the throughput decreases drastically. We observe that the performance of SR improves with increase in window size. As the window size is increased, more packets can be sent from sender and only the lost packets need to be retransmitted.

IV. Conclusion of Experiment 2:



Conclusion: We can see that throughput of SR is better than GBN and ABT for different window sizes and loss probabilities. Window size does not affect ABT, but ABT is a stop and wait protocol due to which the throughput is low. In case of GBN, whole window needs to be retransmitted when a packet is lost. So, throughput decreases with increase in loss and window size. In SR, only lost packets need to be retransmitted. So, SR perform better than the two protocols.