EECE 7374

Fundamentals of Computer Networks

Programming Assignment 2

Academic integrity - We have read and understood the course academic integrity policy.

Team Members:

1) **GOUTHAM GANESH RAJAN (NU ID - 002728596)**
2) **DHIVYA KUMARAGURUPARAN (NU ID - 002665287)**

ALTERNATING BIT PROTOCOL:

- Alternating bit protocol is a type of protocol that operates in the data link layer.
- ABT retransmits the lost or corrupted messages using a method called “First in First Out”.
- In alternating bit protocol, the window size is 1 and the sequence number varies between 0 and 1.
- Alternating bit protocol is a “Connectionless protocol”. It is otherwise called “Stop and Wait Protocol”.

EXECUTION

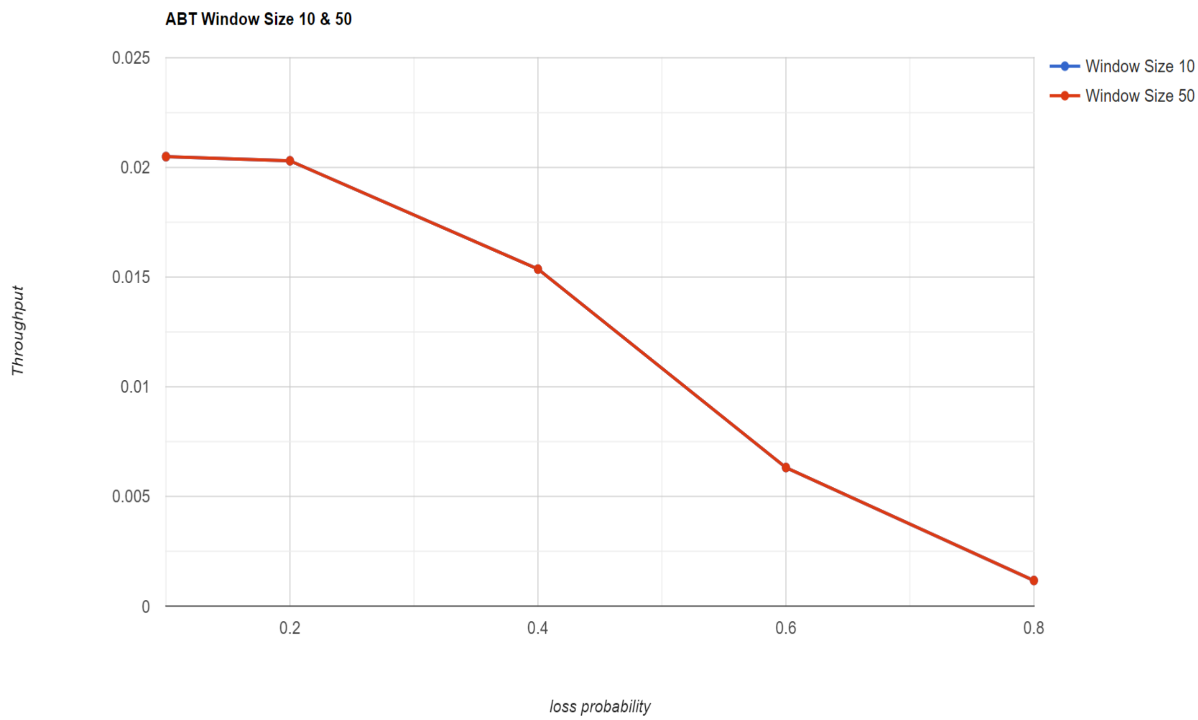
- Based on the Finite State Machine diagram, we have programmed the sender and receiver.
- So, the sender (let’s say A) transmits a message to the receiver (let’s say B).
- Once sender A transmits the message with the sequence number 0 or 1, it will wait for the acknowledgement for the transmitted message.
- Sender A will buffer the next packet until it receives an acknowledgement from the receiver end.
- For ABT at any instance of time, there can be only one unacknowledged packet in its pipeline.
- Now coming to the receiver end, it inspects the packet checksum and based on the received packets it will decide either to give positive or negative acknowledgements.

TIMEOUT SCHEME:

- A constant time has been employed in Alternating Bit Protocol. Following a review of the throughput behavior for various timeout values.

THROUGHPUT ANALYSIS: The below image depicts the throughput for the Alternating Bit Protocol for various loss probabilities.

LOSS PROBABILITY	WINDOW SIZE (10)	WINDOW SIZE (50)
0.1	0.020492	0.020492
0.2	0. 020303	0. 020303
0.4	0. 015364	0. 015364
0.6	0. 006318	0. 006318
0.8	0. 001168	0. 001168



OBSERVATION: According to the figure above, the throughput of ABT decreases as the loss probability increases. This could be because of ABT repeatedly sending the same packet until it is correctly received.

GO-BACK-N:

- This protocol includes a sliding window with a Window Size of N. N packets are sent by the sender to the receiver. The sender has no idea what happens at the other end, and vice versa.
- The sender receives cumulative ACKs from the receiver.
- The sender keeps a timer for the oldest unacknowledged data packet, and the total number of unacknowledged packets cannot exceed the window size.

EXECUTION:

The implementation is carried out very similarly to how the FSM of GBN is carried out, with the exception that the packets are buffered rather than being dropped

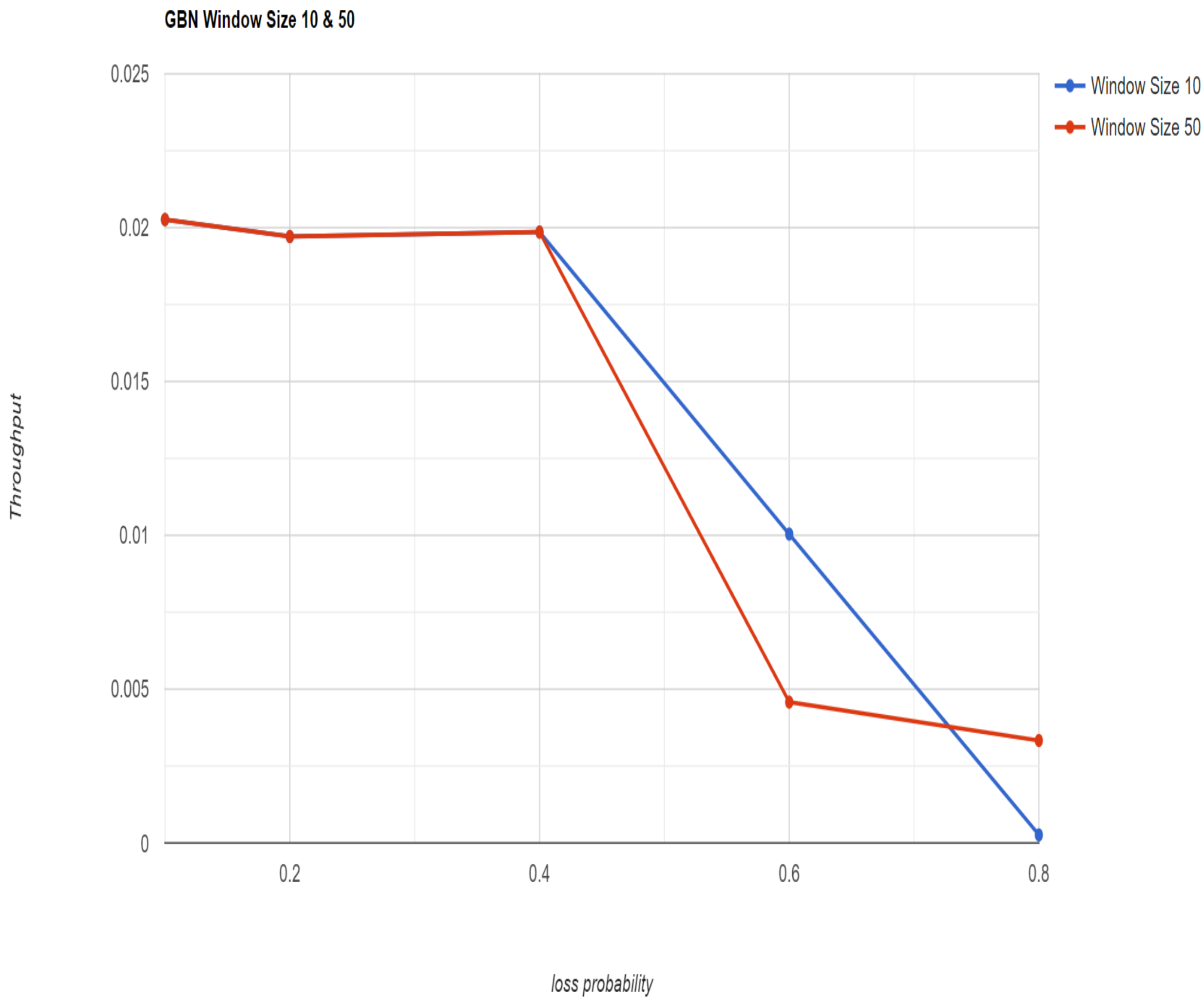
- Sender:
    1. A packet is only transmitted if its size is less than or equal to the window size; otherwise, is buffered.
    2. If timeout happens, all the packets from the current base up to the next available sequence number or up until the window size (from the current base) are sent.
    3. Because cumulative acknowledgements are provided in GBN, whenever we receive one, we first determine whether it is greater than the most recently updated base, and if it is, we add one to that number and then update the base.
- Receiver:
    1. An acknowledgement is sent to the sender, after the sequence number and checksum of the packet are validated after the packet has been received at
    2. Only the correctly received packet that has the greatest sequence number will have an acknowledgement triggered for it.

TIMEOUT SCHEME:

- In GBN, we have made use of an adaptive timer that adjusts itself based on the size of the window.
- After conducting a few tests using several timeout values in an effort to identify the timeout value that provides the best possible results, the timer value was finally settled on.
- If the packet is corrupted or lost, the timer interrupt service is used to resend the data that was not acknowledged.

THROUGHPUT ANALYSIS: The below graph and table shows the throughput for different loss probability

LOSS PROBABILITY	WINDOW SIZE (10)	WINDOW SIZE (50)
0.1	0.020264	0.020264
0.2	0.019711	0.019711
0.4	0.019857	0.019857
0.6	0.010040	0.004577
0.8	0.000259	0.003329



OBSERVATION: As we can observed in the picture that was just presented, when the window size is 10, the throughput is better than when the window size is 50. When the loss probability is 0.6 the throughput of window size 10 is greater than the window size 50. As the loss probability increases, the throughput of both the window sizes decreases But, when the loss probability is less then the throughput is unaffected.

SELECTIVE REPEAT:

- The selective repeat is known as sliding window protocol.
- In selective repeat the retransmission is avoided i.e., in GBN incase if there is an error in a packet it may lead to lots of transmission.

EXECUTION:

- Sender:
  1. When the next available sequence number is in the window, a packet is sent, and a timer is started when the current sequence number equals the base.
  2. The packet is marked acknowledged and the base advances to the following acknowledged sequence if the acknowledgement number of the received packet from B falls within the window.
  3. With each succeeding timeout, the entire timer implementation is handled in the case of SR. The timer is begun in accordance with the timeout system outlined below.
- Receiver:
  1. The packet's sequence number is verified to see if it is intact and falls within the window commencing at the receive base. The packet is then transmitted to the application layer if the sequence number matches what was anticipated, in which case it is placed in the receive buffer and an acknowledgement is given for the packet that was received.
  2. The receive base is raised to the upcoming predicted sequence number.
  3. In the same way as in GBN, the timer interrupt handles retransmission in loss and corruption circumstances. A packet's timeout determines whether it gets retransmitted in SR.

TIMEOUT SCHEME:

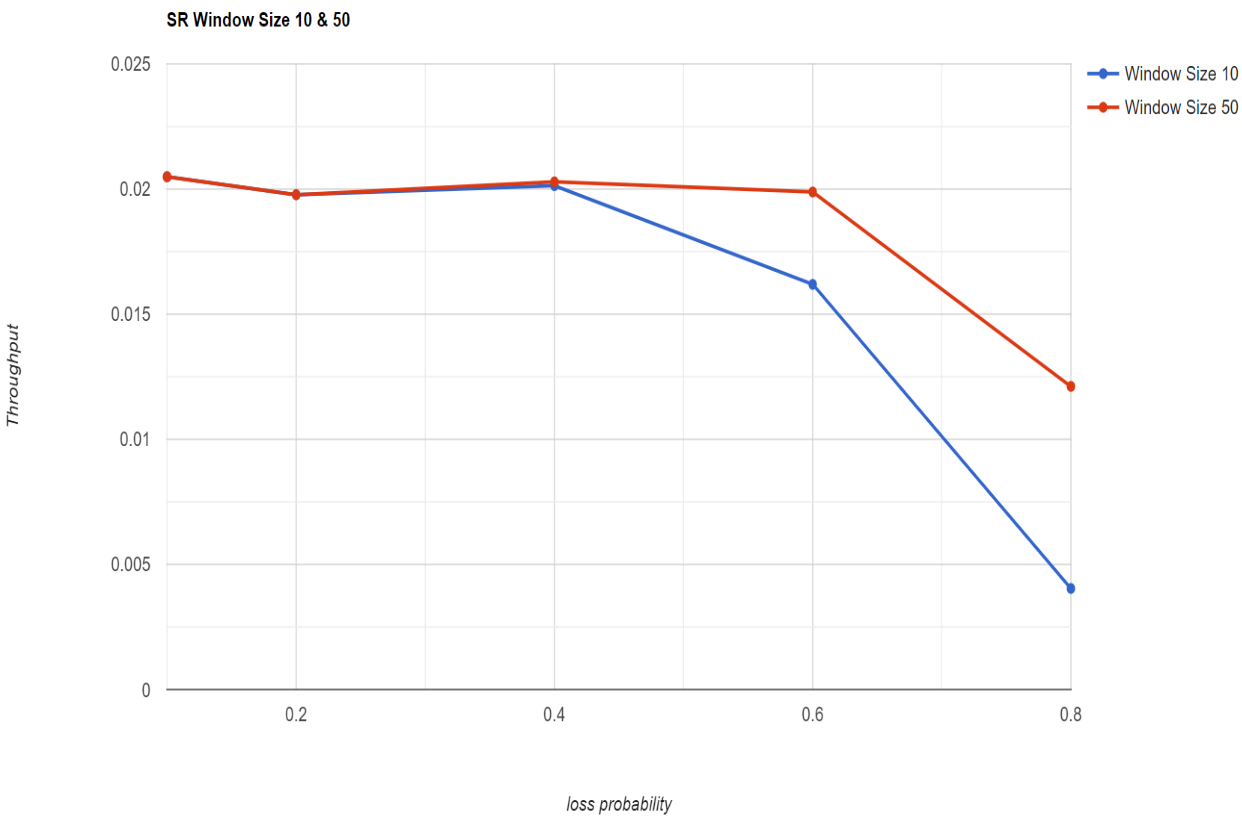
- Unlike GBN, where the timer is only associated with the current window base, we store timers for each individual packet in selective repeat. We also only have one hardware timer, which is utilized to simulate the operation of many logical timers.

EXPLANATION:

- Each packet has a start time associated with it. The difference between the start time of the current packet and the start time of the base packet for the Nth packet is the relative timeout. The start time is the time at which the packet is sent.
- The timer is initially started for the window's base packet, and then, at timeout, it is restarted for the packet with the shortest timeout and no acknowledgement.
- In the event of retransmissions, the packet start time is modified to ensure that the timer for the packet that was delivered earlier than packet N should start.

THROUGHPUT ANALYSIS: The below graph and table shows the throughput for different loss probability

LOSS PROBABILITY	WINDOW SIZE (10)	WINDOW SIZE (50)
0.1	0.020494	0.020494
0.2	0.019775	0.019775
0.4	0.020137	0.020290
0.6	0.016192	0.019884
0.8	0.004039	0.012115



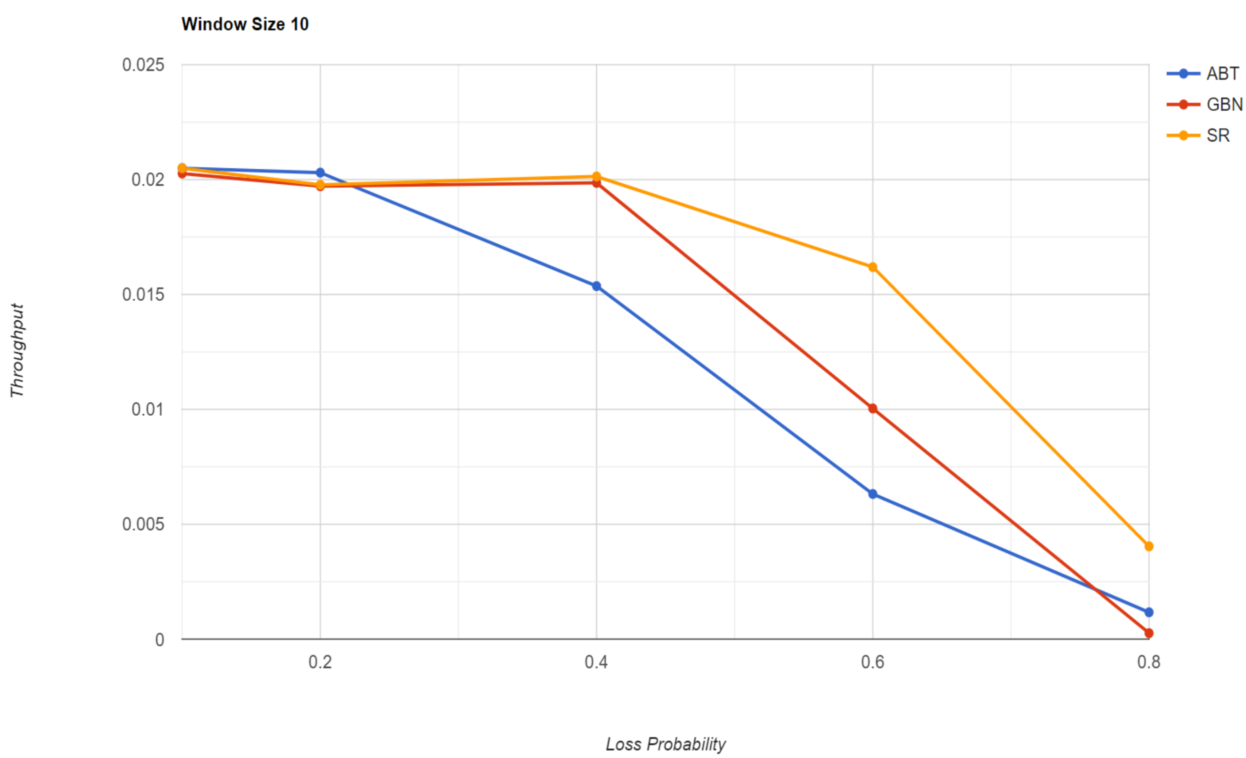
OBSERVATION: For both window sizes, the throughput of SR is nearly identical, and the loss probability has little impact on it. The performance of SR declines when the loss probability is 0.8 and the corruption is 0.2, and it is nearly equivalent for all loss probabilities between 0.1 and 0.6.

EXPERIMENT 1

AIM: 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.

i) When window size= 10

LOSS PROBABILITY	ABT	GBN	SR
0.1	0.020492	0.020264	0.020494
0.2	0. 020303	0.019711	0.019775
0.4	0. 015364	0.019857	0.020137
0.6	0. 006318	0.010040	0.016192
0.8	0. 001168	0.000259	0.004039



OBSERVATION:

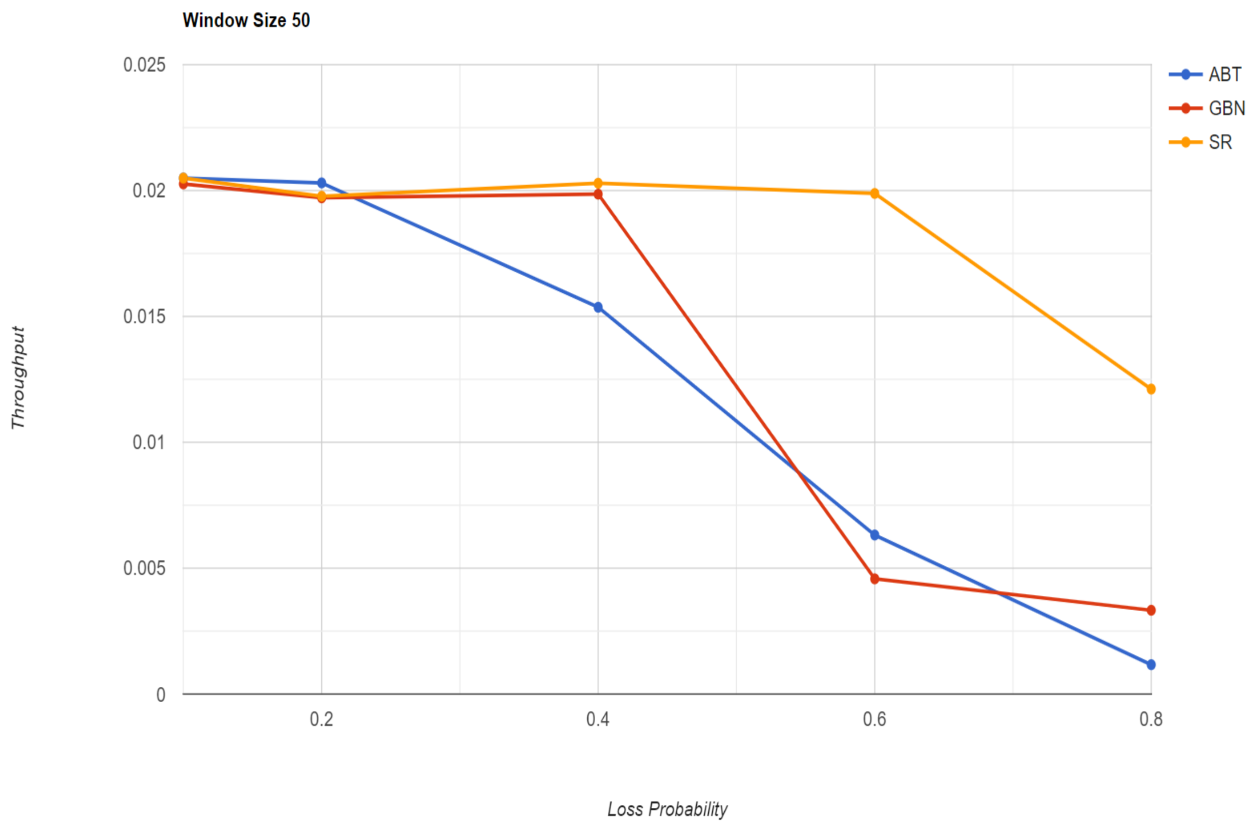
- When the loss probability is 0.1 and 0.2, all the three protocols have similar throughput.
- SR performs better than other protocols when the loss probability increases comparatively.
- GBN and SR have the same throughput for loss probabilities 0.1, 0.2 and 0.4 and GBN throughput value decreases linearly with increase in the loss probability.

INFERENCE:

- SR throughput value decreases but still performs better than the other protocols.

ii) When window size= 50

LOSS PROBABILITY	ABT	GBN	SR
0.1	0.020492	0.020264	0.020494
0.2	0. 020303	0.019711	0.019775
0.4	0. 015364	0.019857	0.020290
0.6	0. 006318	0.004577	0.019884
0.8	0. 001168	0.003329	0.012115



OBSERVATION:

- When the loss probability is 0.1, all the three protocols have similar throughput.
- Comparatively, SR performs better than other protocols when the loss probability increases
- GBN and SR have the same throughput for loss probabilities 0.1, 0.2 and 0.4 and GBN throughput value decreases and is the lowest at 0.8
- Thus, loss probability affects the performance for GBN.
- ABT throughput efficiency is the lowest.

INFERENCE:

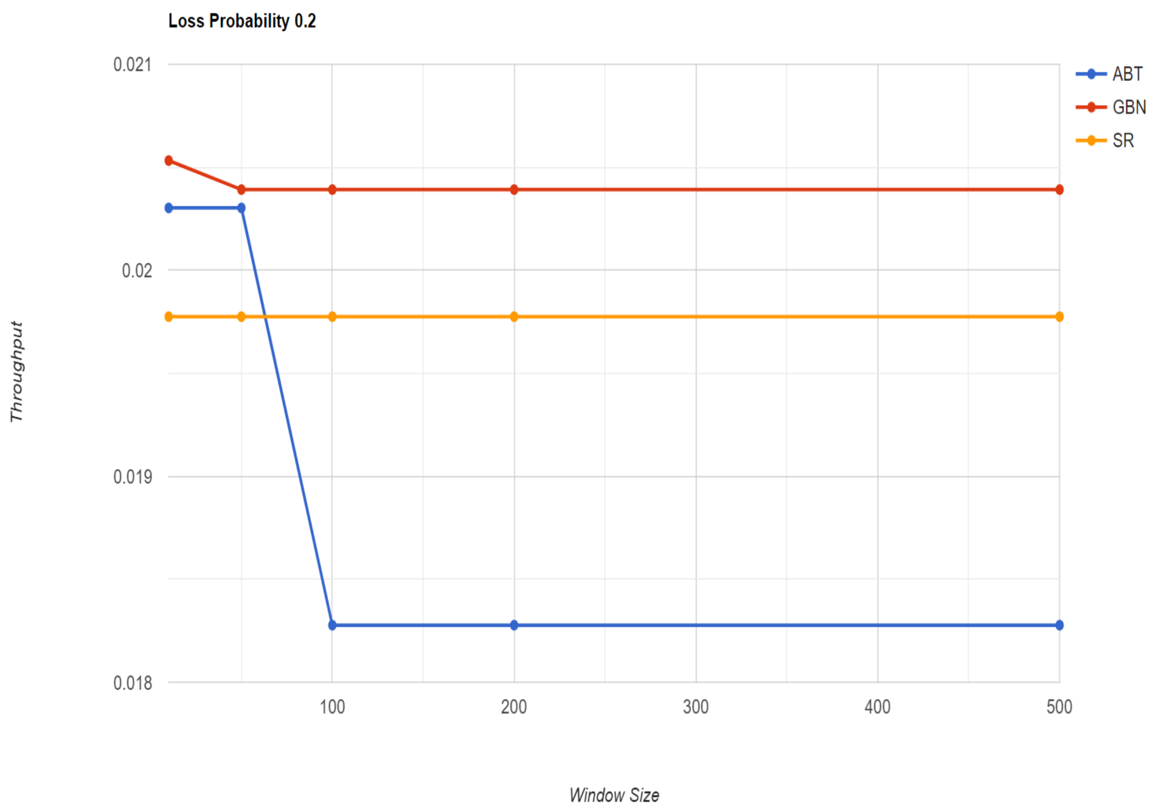
- The SR throughput value decreases but still performs better than the other protocols. Thus, the throughput value is affected by the loss probability, but not as much as the other protocols.

EXPERIMENT 2

AIM: 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.

iii) When loss probability =0.2

WINDOW SIZE	ABT	GBN	SR
10	0.020303	0.020533	0.019775
50	0.020303	0.020392	0.019775
100	0.018277	0.020392	0.019775
200	0.018277	0.020392	0.019775
500	0.018277	0.020392	0.019775



OBSERVATION:

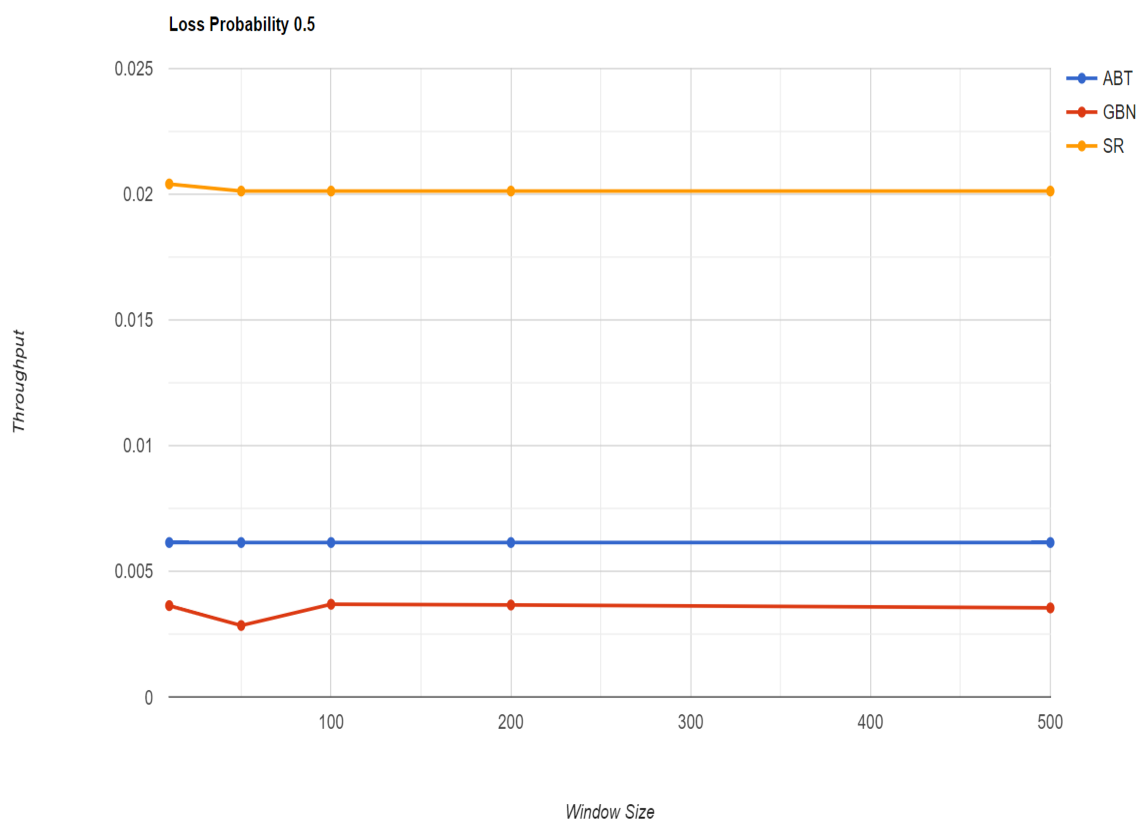
- We can see that the throughput of GBN and SR are greater than the throughput of ABT.
- For ABT, the throughput decreases with the window size. On the other hand, the throughput for GBN and SR is constant for all the window size.

INFERENCE:

- Thus, for loss probability 0.2, GBN and SR perform well.

iv) When loss probability =0.5

WINDOW SIZE	ABT	GBN	SR
10	0.006144	0.003635	0.020410
50	0.006144	0.002843	0.020127
100	0.006144	0.003687	0.020127
200	0.006144	0.003658	0.020127
500	0.006144	0.003543	0.020127



OBSERVATION:

- We can see that the throughput efficiency of ABT outperforms GBN with the increase in the window size, but it is low when compared to SR.
- SR protocol has the highest throughput when compared to the other two protocols.
- The throughput for GBN is the lowest when the window size is 50.

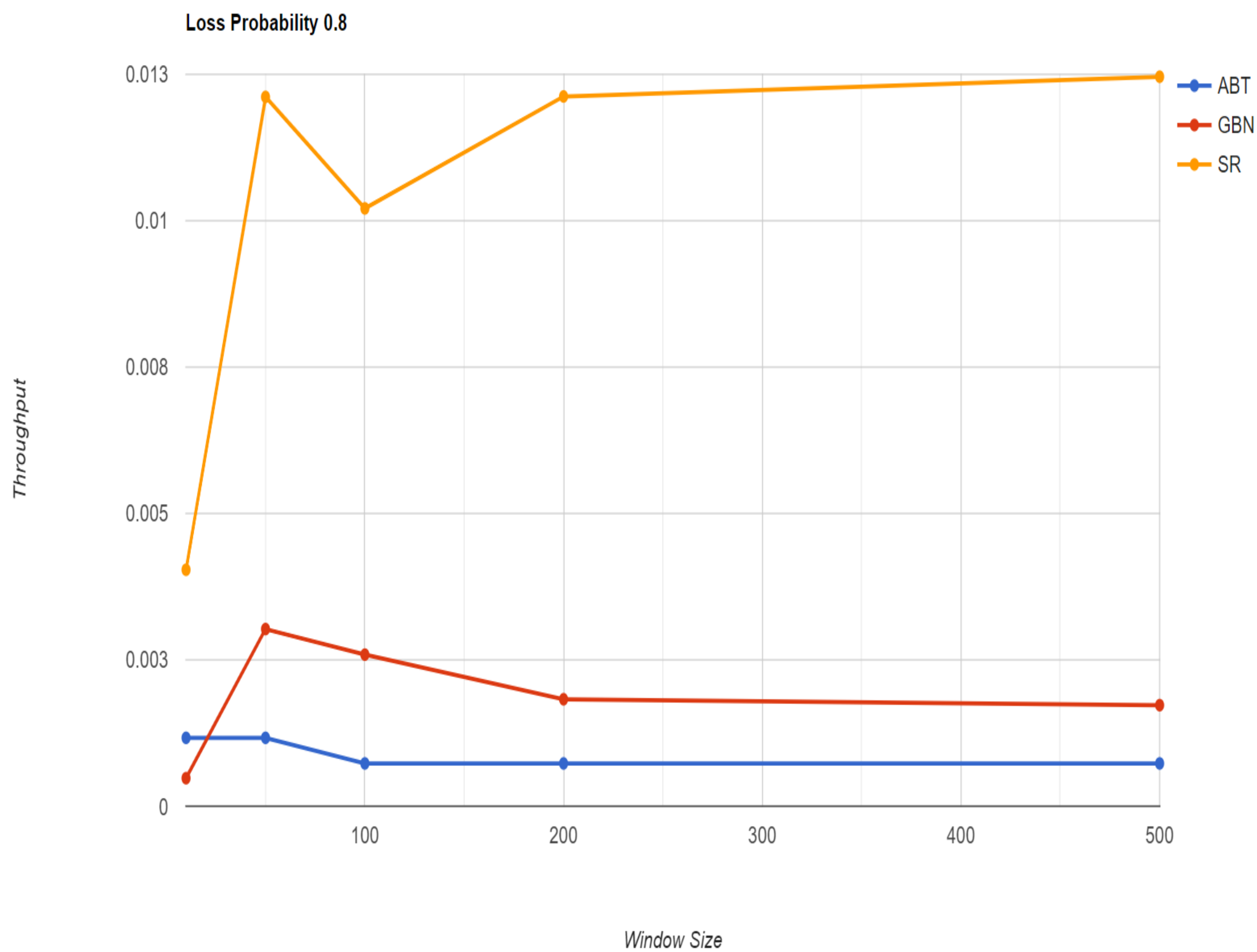
- The throughput of ABT is the same for all the window sizes. Such that it is unaffected by the increase in the window sizes.

INFERENCE:

- Thus, for loss probability 0.5, SR performs well, unaffected by the window sizes. It has maximum throughput when the window size is 10.

v) When loss probability =0.8

WINDOW SIZE	ABT	GBN	SR
10	0.001168	0.000479	0.004039
50	0.001168	0.003026	0.012115
100	0.000731	0.002589	0.010211
200	0.000731	0.001827	0.012123
500	0.000731	0.001726	0.012460



OBSERVATION:

- We can see that the throughput efficiency of SR is the maximum for all window sizes.
- ABT has the lowest throughput efficiency and has the maximum when the window sizes are 10 and 50.
- The throughput for GBN is greater than ABT the throughput is maximum when the window size is 50.
- SR has a linear increase in its throughput as the window sizes increase.

INFERENCE:

- Thus, for loss probability 0.8, SR performs well, when compared to the other two protocols.

RESULT:

- SR consistently delivers the best performance across.
- ABT is efficient when the loss probability is less
- The throughput of GBN drops as both the window size and the loss probability increase.