Spring Semester, 2023

CSE 4/589
Modern Networking Concepts

Programming Assignment 2

Authors:

Sayli Umesh Nadkar (UBIT: sayliume)

Akash Chaitanya Yadav (UBIT: ayadav4)

CSE 489/589

Programming Assignment 1 Stage 1 Report

Reliable Transport Protocols

Notes: (IMPORTANT)

- ➔ One of your group members select <File> - <Make a copy> to make a copy of this report for your group, and share that Google Doc copy with your teammates so that they can also edit it.
- ➔ Report your work in each section. Describe the method you used, the obstacles you met, how you solved them, and the results. You can take screenshots at key points. There are NO hard requirements for your description.
- ➔ For a certain test, if you successfully implemented it, **take a screenshot of the result from the grader as required in section 5 (required).** You can just provide the overall result for each test.
- ➔ For a certain test, if you tried but failed to implement it, properly describe your work. We will partially grade it based on the work you did.
- ➔ Do **NOT** claim anything you didn't implement. If you didn't try on a certain protocol or test, leave that section blank. We will run your code, and if it does not match the work you claimed, you and your group won't get any partial grade score for this WHOLE assignment.
- ➔ There will be **15.0** points for this report. These are NOT bonus points and will be given based on the completion of the analysis part (section 6.1).
- ➔ If you decide not to attempt the analysis part (section 6.1) of the assignment, you will still NEED to submit this report with the requirements stated in section 6.
- ➔ Under **NO** circumstances may you rely on the work of your peers, including but not limited to GitHub repositories or code submissions from previous academic terms.
- ➔ All the analysis results in section 6 should come from one of the provided hosts, NOT on your local machine (see section 3.1 in the handout).
- ➔ The maximum score for PA 2: 85 + 15 = 100

1. Academic Integrity Policy Statement

I have read and understood the course's academic integrity policy.

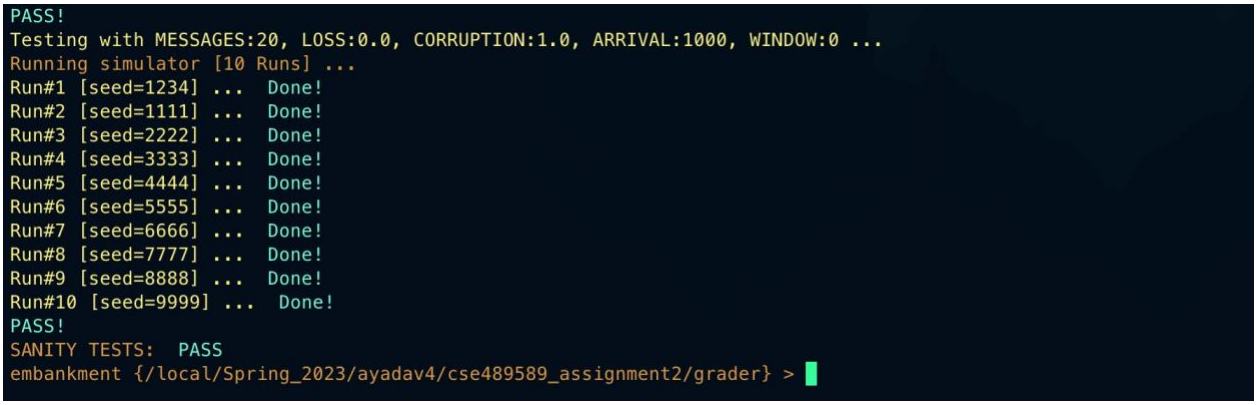
2. Group and Contributions

- Name of member 1:
  - **Sayli Umesh Nadkar** (UBIT: sayliume)
  - Contributions: Protocol analysis (GBR and SR) and code implementation
- Name of member 2:
  - **Akash Chaitanya Yadav** (UBIT: ayadav4)
  - Contributions: Protocol analysis (ABT), Documentation and Experiments.

3. Sanity Tests

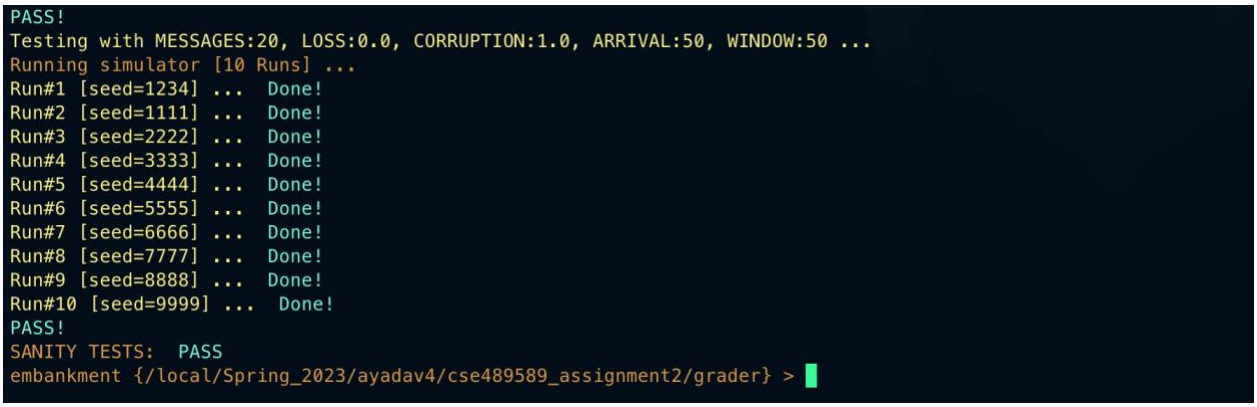
[2.0] ABT (Alternating-Bit Protocol)

Grader screenshot:



[5.0] GBN (Go-Back-N Protocol)

Grader screenshot:



[8.0] SR (Selective-Repeat Protocol)

Grader screenshot:

```
PASS!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:1.0, ARRIVAL:50, WINDOW:50 ...
Running simulator [10 Runs] ...
Run#1 [seed=1234] ...  Done!
Run#2 [seed=1111] ...  Done!
Run#3 [seed=2222] ...  Done!
Run#4 [seed=3333] ...  Done!
Run#5 [seed=4444] ...  Done!
Run#6 [seed=5555] ...  Done!
Run#7 [seed=6666] ...  Done!
Run#8 [seed=7777] ...  Done!
Run#9 [seed=8888] ...  Done!
Run#10 [seed=9999] ...  Done!
PASS!
SANITY TESTS:  PASS
embankment {/local/Spring_2023/ayadav4/cse489589_assignment2/grader} >
```

4. Basic Tests

[5.0] ABT (Alternating-Bit Protocol)

Grader screenshot:

```
PASS!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.8, ARRIVAL:1000, WINDOW:0 ...
Running simulator [10 Runs] ...
Run#1 [seed=1234] ...  Done!
Run#2 [seed=1111] ...  Done!
Run#3 [seed=2222] ...  Done!
Run#4 [seed=3333] ...  Done!
Run#5 [seed=4444] ...  Done!
Run#6 [seed=5555] ...  Done!
Run#7 [seed=6666] ...  Done!
Run#8 [seed=7777] ...  Done!
Run#9 [seed=8888] ...  Done!
Run#10 [seed=9999] ...  Done!
PASS!
BASIC TESTS:  PASS
embankment {/local/Spring_2023/ayadav4/cse489589_assignment2/grader} >
```

Description:

[12.0] GBN (Go-Back-N Protocol)

Grader screenshot:

```
PASS!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:1.0, ARRIVAL:50, WINDOW:50 ...
Running simulator [10 Runs] ...
Run#1 [seed=1234] ...  Done!
Run#2 [seed=1111] ...  Done!
Run#3 [seed=2222] ...  Done!
Run#4 [seed=3333] ...  Done!
Run#5 [seed=4444] ...  Done!
Run#6 [seed=5555] ...  Done!
Run#7 [seed=6666] ...  Done!
Run#8 [seed=7777] ...  Done!
Run#9 [seed=8888] ...  Done!
Run#10 [seed=9999] ...  Done!
PASS!
SANITY TESTS:  PASS
embankment {/local/Spring_2023/ayadav4/cse489589_assignment2/grader} >
```

[18.0] SR (Selective-Repeat Protocol)

Grader screenshot:

```
PASS!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.8, ARRIVAL:50, WINDOW:50 ...
Running simulator [10 Runs] ...
Run#1 [seed=1234] ...  Done!
Run#2 [seed=1111] ...  Done!
Run#3 [seed=2222] ...  Done!
Run#4 [seed=3333] ...  Done!
Run#5 [seed=4444] ...  Done!
Run#6 [seed=5555] ...  Done!
Run#7 [seed=6666] ...  Done!
Run#8 [seed=7777] ...  Done!
Run#9 [seed=8888] ...  Done!
Run#10 [seed=9999] ...  Done!
PASS!
BASIC TESTS:  PASS
embankment {/local/Spring_2023/ayadav4/cse489589_assignment2/grader} >
```

5. Advanced Tests

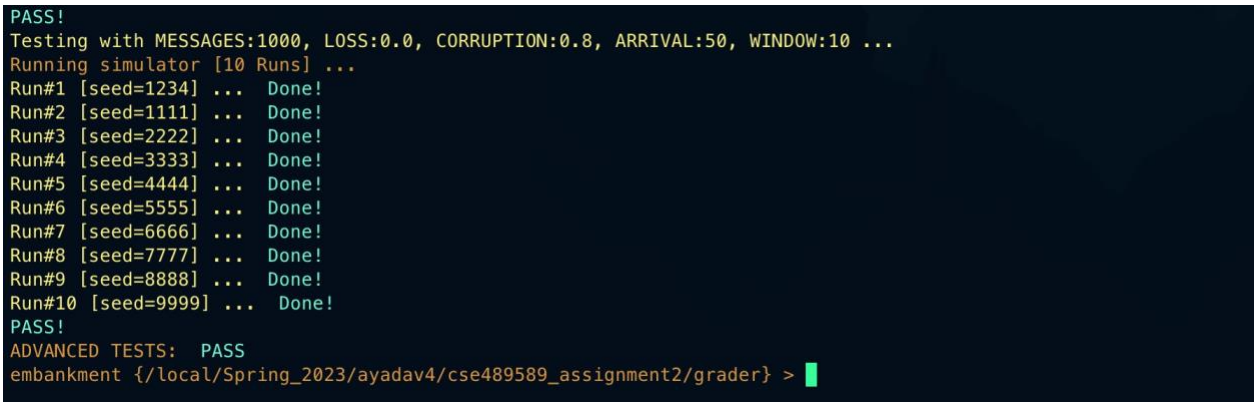
[5.0] ABT (Alternating-Bit Protocol)

Grader screenshot:

```
PASS!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.8, ARRIVAL:50, WINDOW:0 ...
Running simulator [10 Runs] ...
Run#1 [seed=1234] ...  Done!
Run#2 [seed=1111] ...  Done!
Run#3 [seed=2222] ...  Done!
Run#4 [seed=3333] ...  Done!
Run#5 [seed=4444] ...  Done!
Run#6 [seed=5555] ...  Done!
Run#7 [seed=6666] ...  Done!
Run#8 [seed=7777] ...  Done!
Run#9 [seed=8888] ...  Done!
Run#10 [seed=9999] ...  Done!
PASS!
ADVANCED TESTS:  PASS
embankment {/local/Spring_2023/ayadav4/cse489589_assignment2/grader} >
```

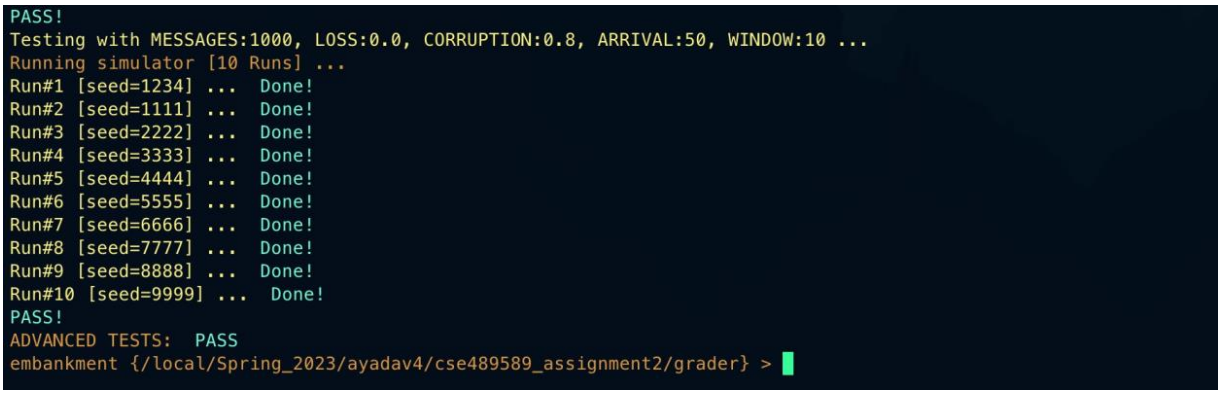
[10.0] GBN (Go-Back-N Protocol)

Grader screenshot:



[20.0] SR (Selective-Repeat Protocol)

Grader screenshot:



6. ANALYSIS & REPORT **[15.0]**

(We expect you to use graphs to show your results for each of the experiments in 6.1 and then write down your observations. Further, your report, at the very least, should answer questions like: What variations did you expect for throughput by changing those parameters and why? Do you agree with your measurements; if not, then why?)

In each of the following 2 experiments, run each of your protocols with a total number of 1000 messages to be sent by entity A, a mean time of 50 between message arrivals (from A's Layer 5), and a corruption probability of 0.2.

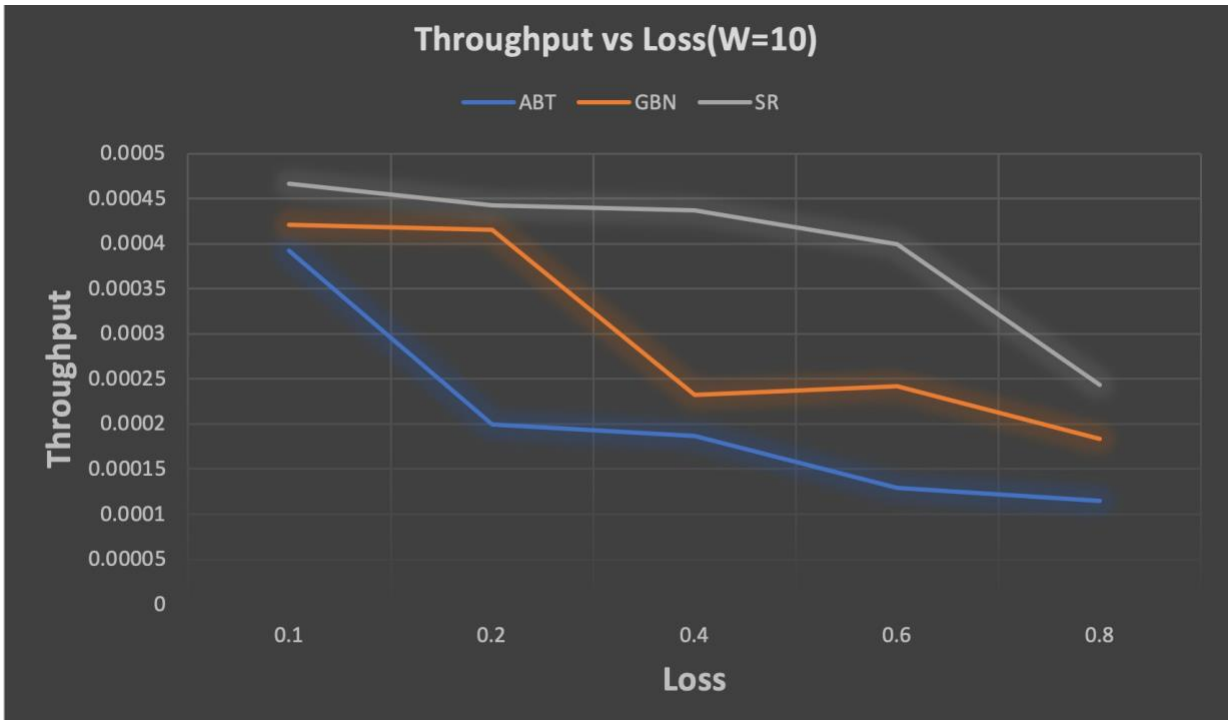
Experiment 1

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.

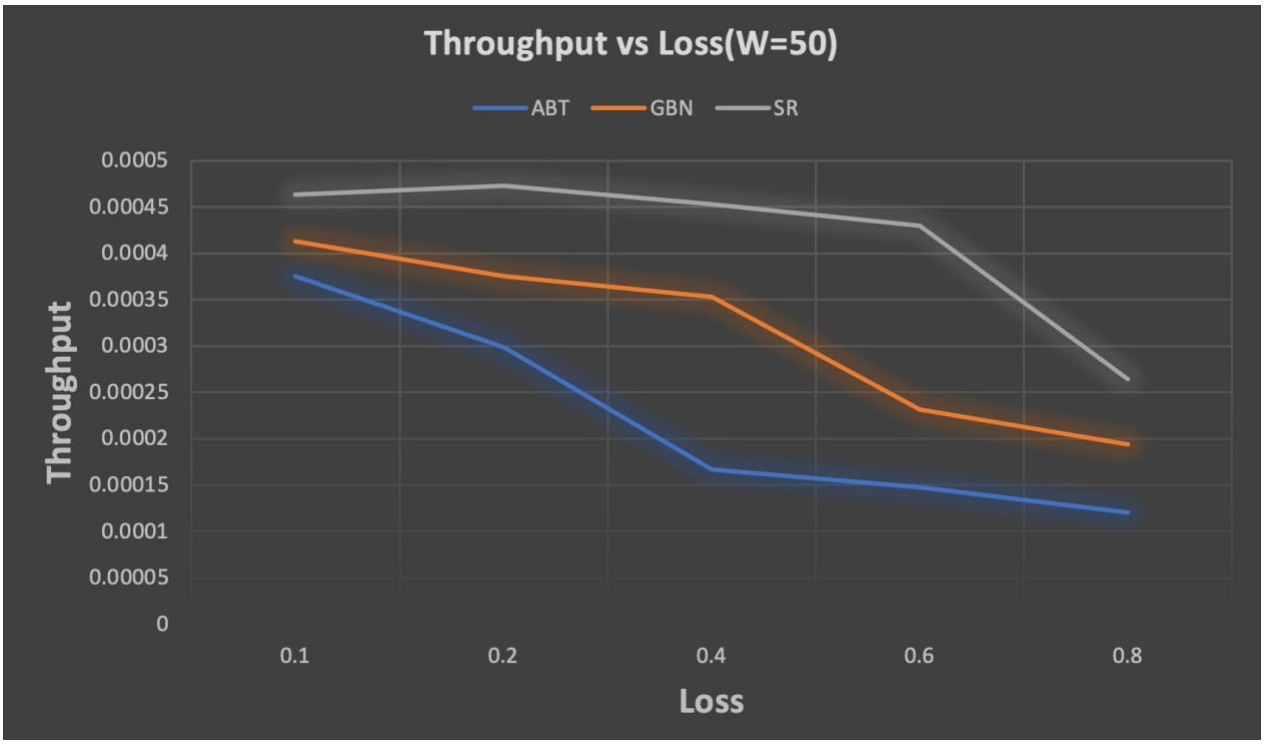
Expected Graphs

- Window size: 10; X-axis: Loss probability; Y-axis: Throughput (ABT, GBN and SR) in one graph/plot.
- Window size: 50; X-axis: Loss probability; Y-axis: Throughput (ABT, GBN and SR) in one graph/plot.

Below is the observation for the three protocols at Window size 10. The graph is provided below:



Below is the observation for the three protocols at Window Size 50 with loss probabilities: {0.1, 0.2, 0.4, 0.6, 0.8}. The graph is provided below:



Upon comparing the given throughput and window size values for ABT, GBN and SR, the following observations can be made:

1. For both window sizes, Selective Repeat (SR) has the highest throughput compared to ABT and GBN.
2. As the loss probability increased, the throughput for all protocols decreased.
3. For both window sizes, as the loss probability increased, the performance of GBN decreased more rapidly compared to ABT and SR.
4. For window size 50, the throughput of GBN decreased compared to window size 10, while the performance of SR remained the same.
5. ABT's throughput was poor for both window sizes, indicating that the transmissions were constantly timing out with loss of packets.
6. Another observation is that the throughput of GBN decreased for window size 50, while the performance of SR remained the same. This could be due to the reason that SR retransmits only those frames that are suspected to be lost or damaged.

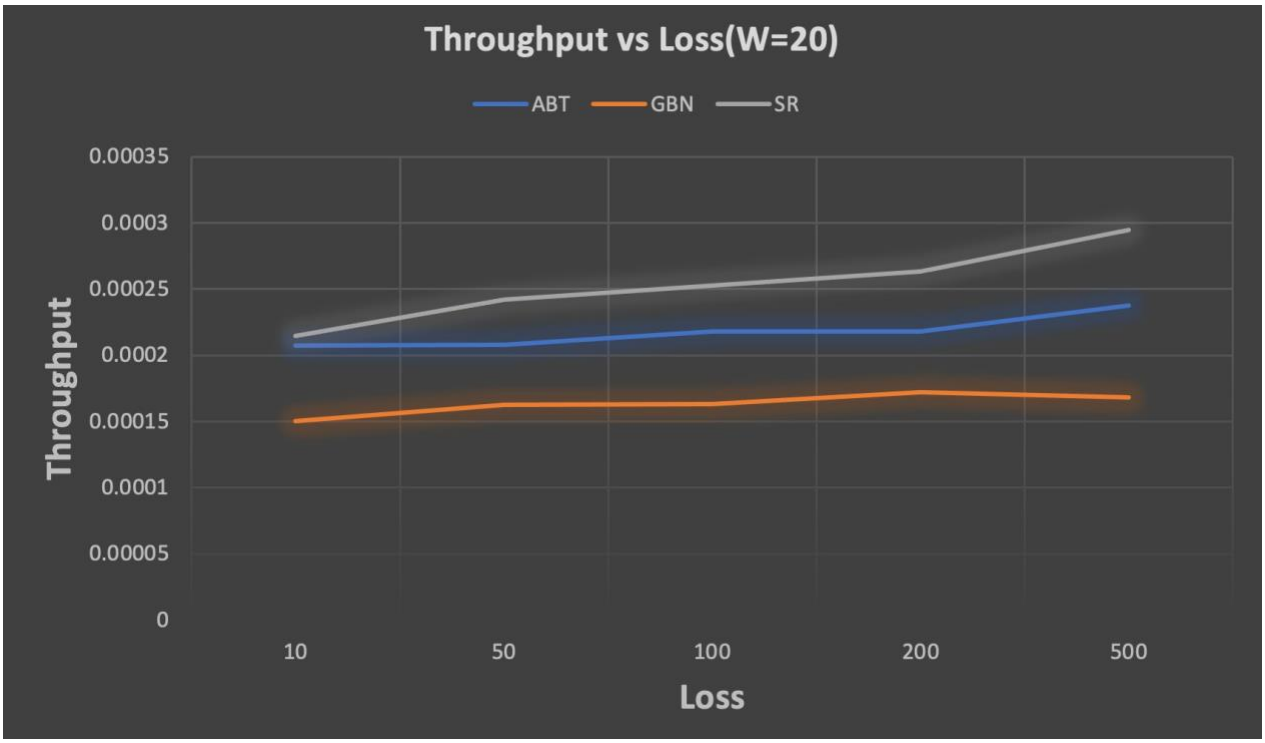
Experiment 2

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.

Expected Graphs

- Loss probability: 0.2; X-axis: Window size; Y-axis: Throughput (ABT, GBN and SR) in one graph/plot.
- Loss probability: 0.5; X-axis: Window size; Y-axis: Throughput (ABT, GBN and SR) in one graph/plot.
- Loss probability: 0.8; X-axis: Window size; Y-axis: Throughput (ABT, GBN and SR) in one graph/plot.

Below is the observation for the three protocols at 0.2 (20%) loss probability. The graph is provided below:

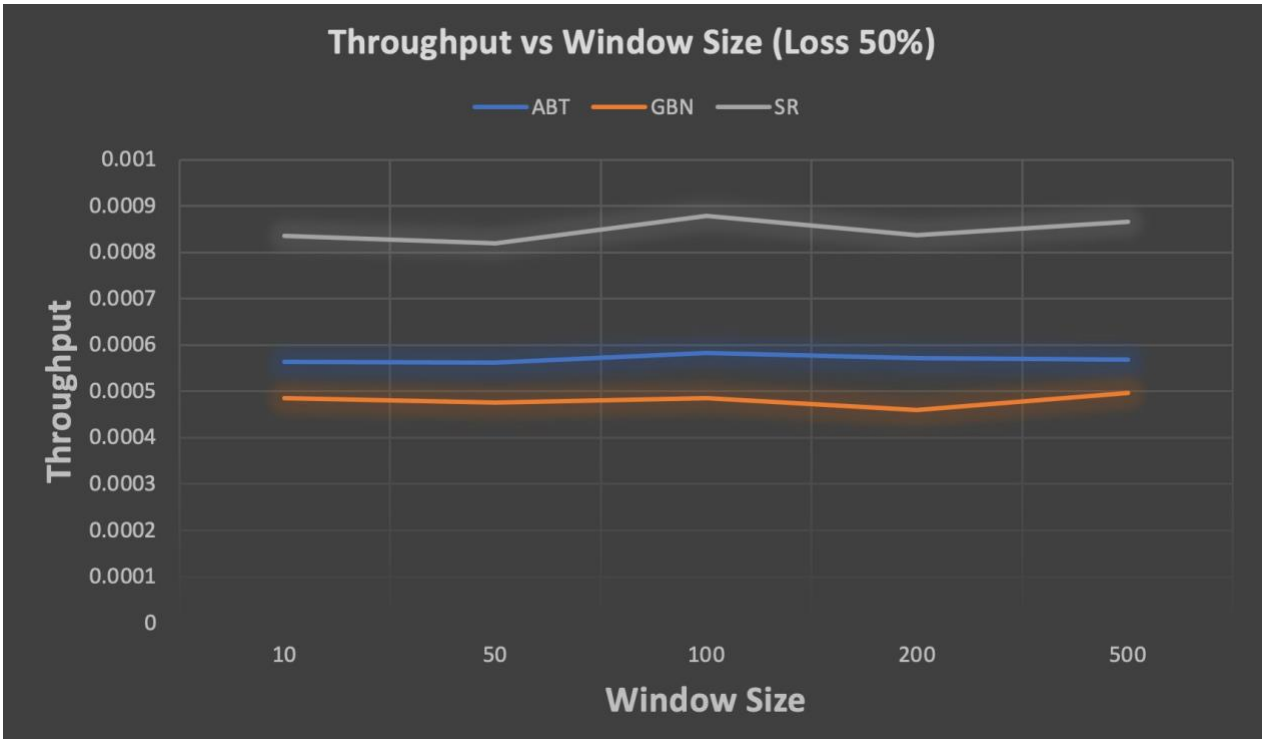


Below are the data points inferenced from the above readings:

1. The throughput values of all three protocols (ABT, GBN, SR) increase with an increase in the window size.
2. At the same window size, the highest throughput is observed for the SR protocol, followed by ABT and GBN.
3. At window size 10, the throughput value of ABT is higher than GBN but at window sizes 50, 100, 200, and 500, the throughput value of GBN is higher than ABT.
4. The throughput values of all three protocols increase significantly between window sizes 10 and 50, but the increase is not as significant for window sizes beyond 50.

5. The highest throughput value observed is for SR protocol at window size 500 and the lowest throughput value is for GBN protocol at window size 10.
6. Across different window sizes, the throughput values of the protocols are not constant, which might indicate that window size alone does affect the throughput values.

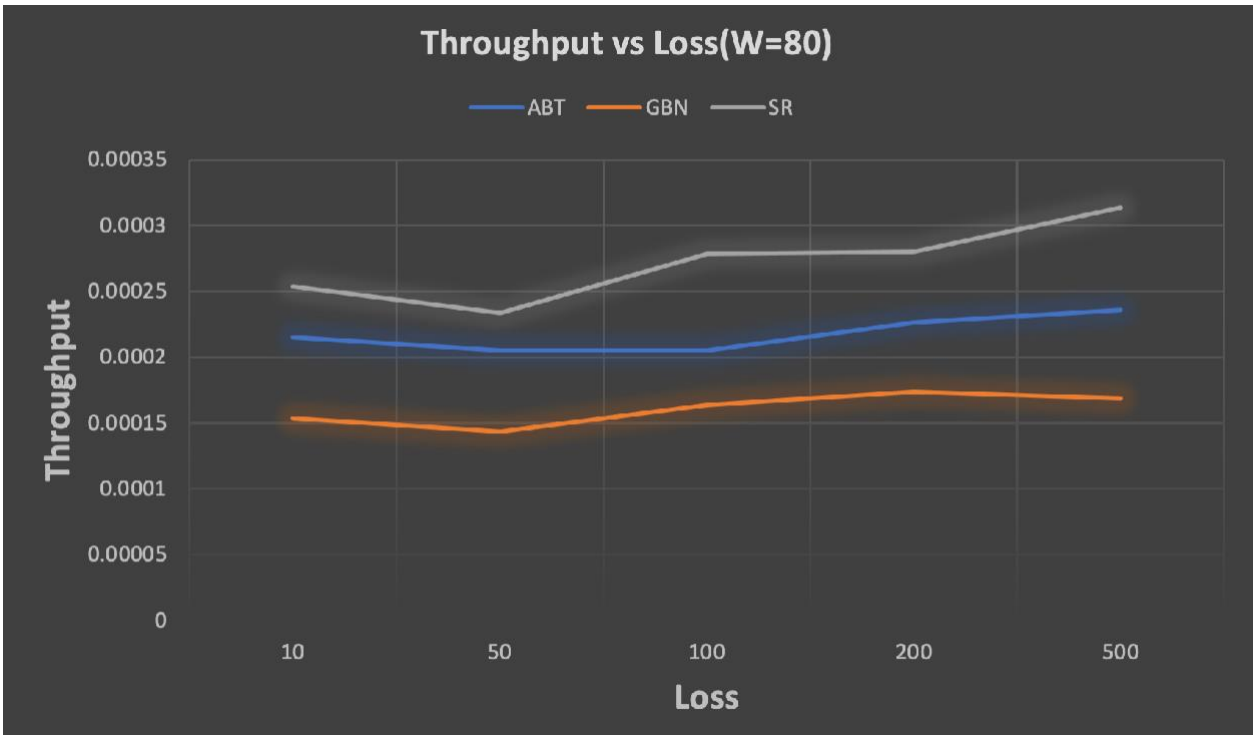
Below is the observation for the three protocols at 0.5 (50%) loss probability. The graph is provided below:



Below are the data points inferenced from the above readings:

1. The throughput values for ABT and SR protocols are almost constant across different window sizes, whereas for GBN it shows a slight decrease as window size increases.
2. The highest throughput value is observed for SR protocol for all the window sizes, followed by ABT and then GBN.
3. The difference in throughput values between the three protocols is not significant.
4. Overall, the window size does not seem to have a major impact on the throughput values for the given set of loss probability values.
5. Selective Repeat (SR) protocol has the highest throughput values across all window sizes, which is expected as SR could selectively retransmit lost packets rather than retransmitting the entire window as in GBN protocol.
6. Increase in loss rate affects the throughput of ABT and GBN protocols more negatively than SR. This could be due to the retransmission of packets in ABT and GBN, leading to more retransmissions and a much higher reduction in throughput compared to SR protocol.

Below is the observation for the three protocols at 0.8 (80%) loss probability. The graph is provided below:



Below are the data points inferenced from the above readings:

1. For Window Sizes 10 and 50, SR has the highest throughput compared to ABT and GBN.
2. For Window Size 100, SR has the highest throughput followed by ABT and then GBN.
3. For Window Sizes 200 and 500, SR has the highest throughput followed by GBN and then ABT.
4. The performance of ABT and GBN are mostly comparable in terms of throughput performance.
5. These observations could be explained by the fact that if the error rate is high, i.e. more frames are being damaged and then retransmitting all the frames that arrived after a damaged frame waste a lot of bandwidth. On the other hand, the Selective Repeat (SR) protocol re-transmits only the damaged frame, hence minimum bandwidth is wasted. And with ABT, the sender can send at most one new packet per RTT, therefore performance-wise it is expected to perform worse than SR.