CSE 489/589
Programming Assignment 2 Report
Reliable Transport Protocols

1 - Academic Integrity Policy Statement

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

2 - Group and Contributions

- Name of member 1:
  - UBITName: **msatao**
  - Contributions:
    - ☐ Design of abt protocol and gbn protocol.
    - ☐ Designed logic for SR multiple software timers
    - ☐ Debugged timer interrupt issues with multiple re-transmission.
    - ☐ Plotting of experiments and results for Experiment 1 and Experiment 2. Collective findings summarized for the experiments.

- Name of member 2:
  - UBITName:**dvasprab**
  - Contributions:
    - ☐ Designed and implemented the rdt3.0 (abt protocol) base.
    - ☐ Found multiple re-transmit errors and inefficient transmissions.
    - ☐ Fix implemented by changing the timer_interrupt logic and acknowledgement logic on the sender side.
    - ☐ Fix for ignoring already received / corrupt/ delayed messages at receiver B and duplicate ack's at sender A.
    - ☐ Designed and implemented gbn protocol.
    - ☐ Multiple performance issues due to high retransmissions were fixed in the A_timerinterrupt section of the code.
    - ☐ Performed sanity tests for large window size and changed the timeout value for failing test case of window 500 in GBN.Discarded transmitting unnecessary packets.
    - ☐ Plotting of experiments and results for Experiment 1 and Experiment 2. Collective findings summarized for the experiments.

3 - SANITY Tests

[2.0] ABT

(

```
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > ./sanity_tests -p ./../dvasprab/abt -r ./run_experiments
Testing with MESSAGES:1000, LOSS:0.1, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.2, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.4, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.6, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.8, CORRUPTION:0.0, 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!
```

```
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.1, 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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.2, 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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.4, 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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.6, 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!
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!
```

```
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:1.0, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:1.0, 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!
SANITY TESTS:  PASS
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > date
Thu May  4 08:41:09 EDT 2023
```

[5.0] GBN

(Put screenshots of the grader here…)

```
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > ./sanity_tests -p ./../dvasprab/gbn -r ./run_experiments
Testing with MESSAGES:1000, LOSS:0.1, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.2, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.4, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.6, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
```

```
Testing with MESSAGES:1000, LOSS:0.8, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.1, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.2, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.4, ARRIVAL:50, WINDOW:10 ...
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!
```

```
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.6, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.8, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:1.0, CORRUPTION:0.0, 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!
```

```
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
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > date
Wed May  3 23:12:33 EDT 2023
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} >
```

[8.0] SR

(Put screenshots of the grader here…)

[No further grading for the protocol that fails a SANITY test.]

4 - BASIC Tests

[5.0] ABT

(Put screenshots of the grader here…)

```
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > ./basic_tests -p ./../dvasprab/abt -r ./run_experiments
Testing with MESSAGES:20, LOSS:0.1, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:0.4, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:0.8, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.1, 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!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.4, 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!
```

```
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
```

[12.0] GBN

(Put screenshots of the grader here…)

```
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > ./basic_tests  -p ./../dvasprab/gbn -r ./run_experiments
Testing with MESSAGES:20, LOSS:0.1, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:0.4, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:0.8, CORRUPTION:0.0, 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!
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.1, 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!
```

```
Testing with MESSAGES:20, LOSS:0.0, CORRUPTION:0.4, 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!
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
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > date
Wed May  3 23:13:43 EDT 2023
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} >
```

[18.0] SR

(Put screenshots of the grader here…)

[No further grading for the protocol that fails a BASIC test.]

5 - ADVANCED Tests

[5.0] ABT

(Put screenshots of the grader here…)

```
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > ./advanced_tests -p ./../dvasprab/abt -r ./run_experiments
Testing with MESSAGES:1000, LOSS:0.1, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.2, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.4, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.6, CORRUPTION:0.0, 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!
Testing with MESSAGES:1000, LOSS:0.8, CORRUPTION:0.0, 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!
```

```
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.1, 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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.2, 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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.4, 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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.6, 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!
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
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > date
Thu May  4 08:41:37 EDT 2023
```

[10.0] GBN

(Put screenshots of the grader here…)

```
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > ./advanced_tests  -p ./../dvasprab/gbn -r ./run_experiments
Testing with MESSAGES:1000, LOSS:0.1, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.2, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.4, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.6, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
```

```
Testing with MESSAGES:1000, LOSS:0.8, CORRUPTION:0.0, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.1, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.2, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.4, ARRIVAL:50, WINDOW:10 ...
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!
```

```
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.6, ARRIVAL:50, WINDOW:10 ...
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!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.8, ARRIVAL:50, WINDOW:10 ...
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
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} > date
Wed May  3 23:24:37 EDT 2023
stones {/local/Spring_2023/dvasprab/cse489589_assignment2/grader} >
```

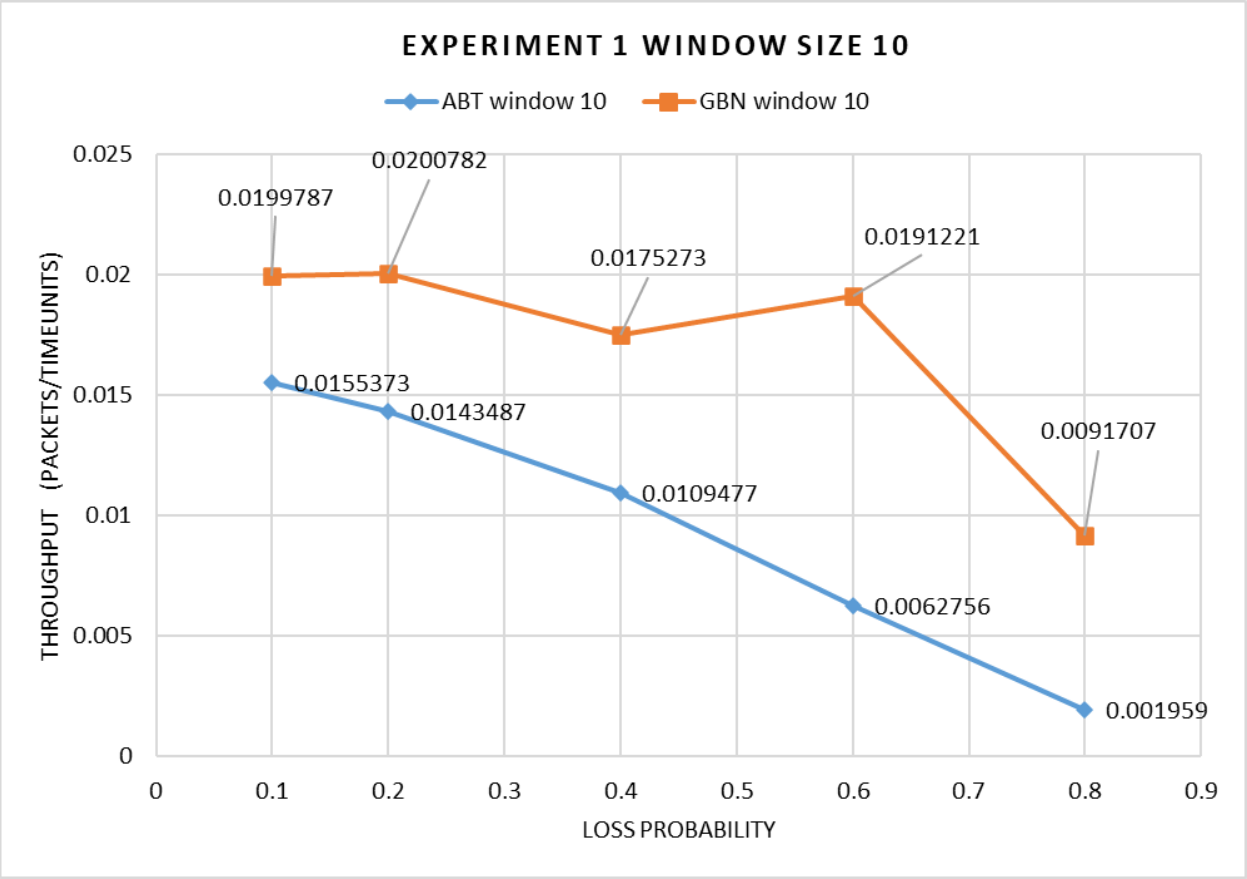
[20.0] SR

(Put screenshots of the grader here…)

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.

EXPERIMENT 1 results :

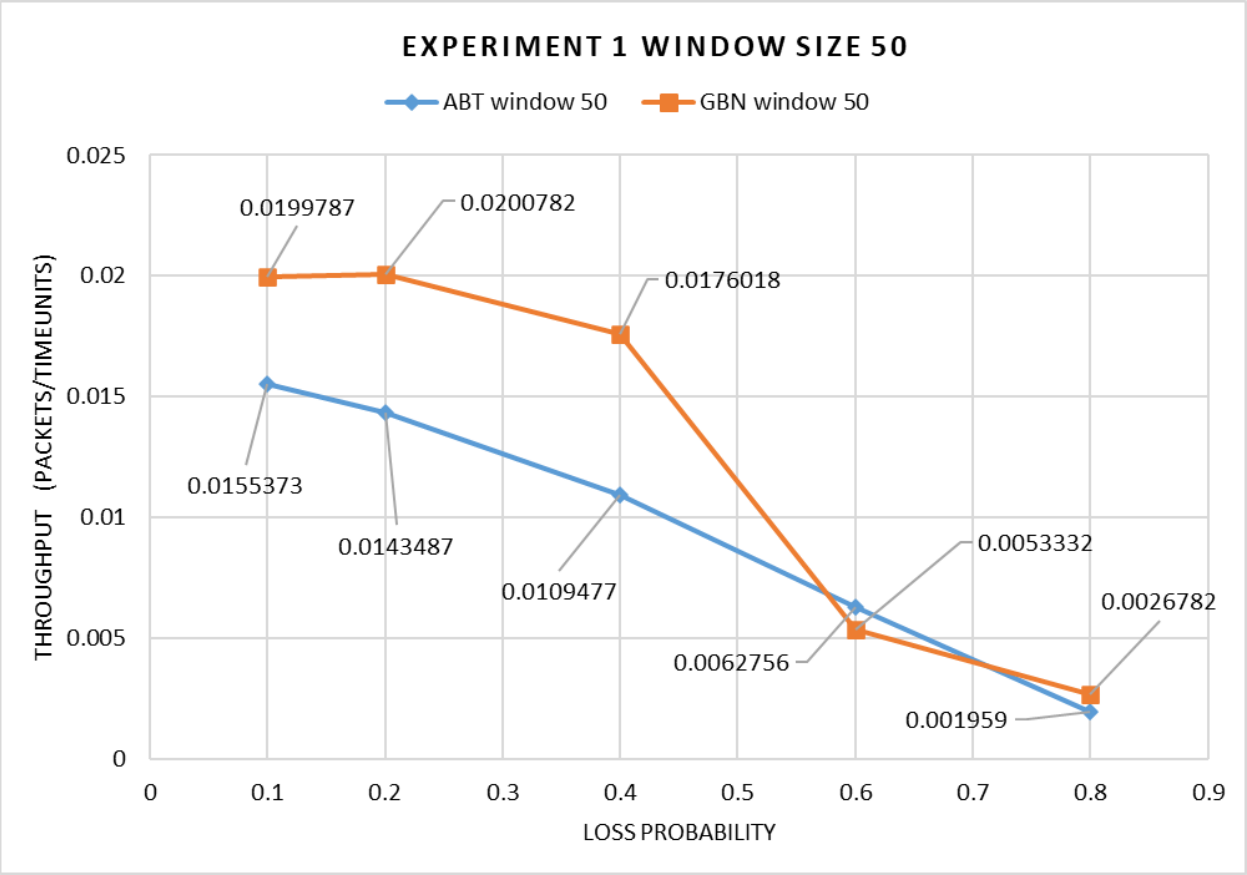


ABT relies on acknowledging each packet from the receiver before sending the next packet. From the chart above there is a trend in that the throughput is decreasing with the increasing loss probability. This evidently means that as more and more packets are lost on theirway from sender A to receiver B, the incoming packets will have to wait before the previous packet is acknowledged. Regardless of whether the packet loss occurs at the sender side or if the acknowledgement is lost from the receiver side, until an ACK is received by sender A for a packet sent, the next packet will not be transmitted. Hence as the loss probability increases , with constant 0.2 corruption probability , we see a linear downward trend of the ABT throughput with respect to loss probability in the channel. The highest throughput achieved is 0.015535 packets/time units .

In contrast, GBN performs much better as compared to ABT since acknowledgements are not awaited within a given window. As more and more packets are transmitted, there is a chance of getting acknowledgements from each of those packets. Moreover, since GBN depends on cumulative acknowledgements, even if the receiver B sends acknowledgement for the last packet and all previous packet ACKS are lost on the way, the sender A is able to figure out that all previous messages were received by the receiver B.

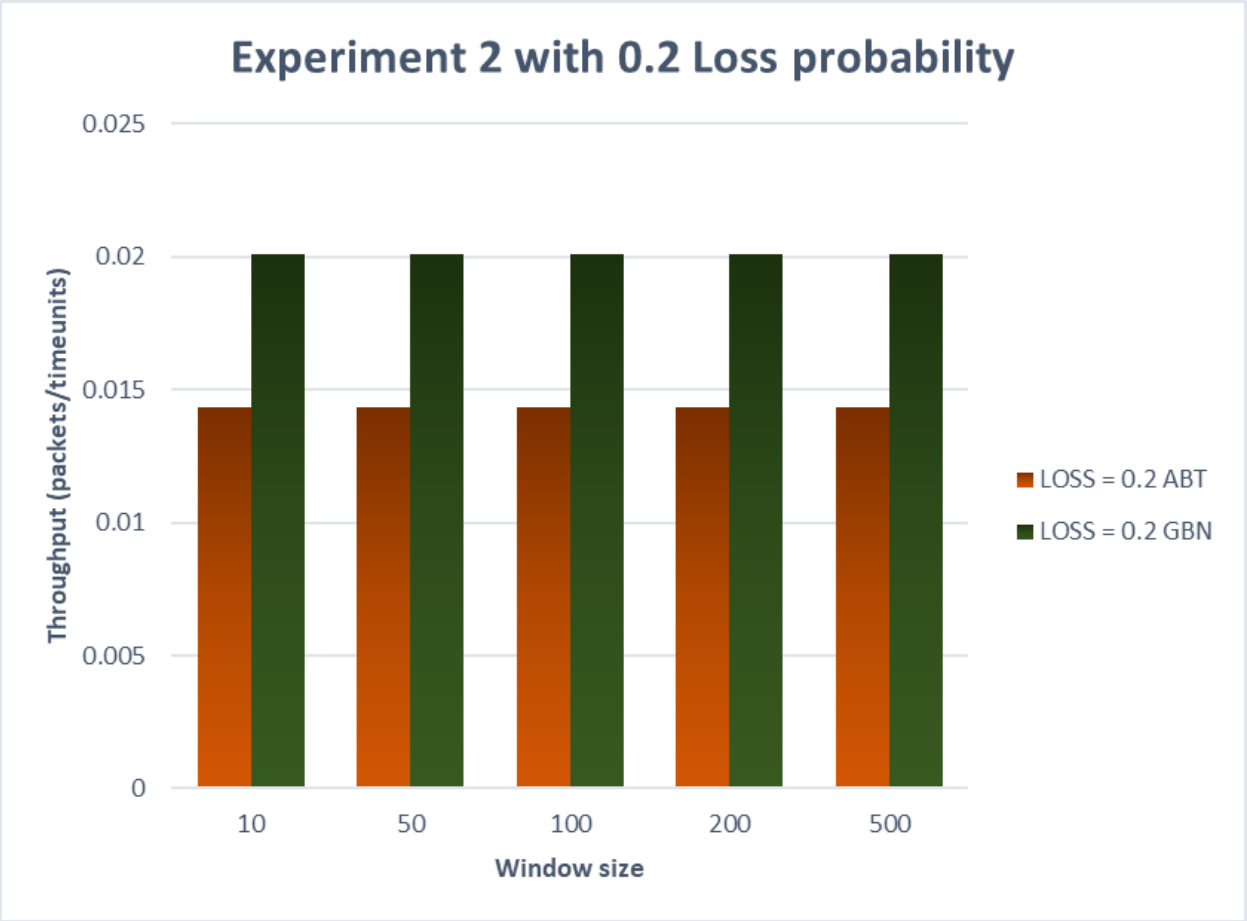
The trend above shows that up until a certain extent, GBN can handle lossy channels, however when the loss exceeds 0.6 , i.e. 6 in 10 transmitted packets are lost, then there is a downward trend. This is expected as higher loss will result in high re-transmissions. Now those

retransmitted packets will have been lost on their way as well, and since the whole window needs to be re-transmitted, this decreases the performance of GBN. As probability of loss increases, probability of lossy retransmits increases, giving rise to the low throughput seen above. There is however an outlier at 0.6 loss probability ,which is something that needs to be looked at.



The above shows the similar findings as discussed above. Here we see that the loss drops significantly at 0.5 onwards. This is due to the bigger window size=50. Bigger window size might be good for lossless channels, but as packets are lost or corrupted, retransmissions of all packets since the last un-ack'd packet will be sent. Regardless of whether the next packets were correctly received or not. For instance, if packet 2 is lost, and packet 3,4,5,6,7 are delivered, but all the ACK's are lost, then all packets since sequence 2, i.e. 2,3,4,5,6,7 will be re-transmitted. This decreases the throughput, and if the retransmissions are lost or delayed, then this will cause further retransmissions resulting in unnecessary duplicate packets at the receiver B. The trend is that as window size increases , the throughput will increases, but this holds good , only if the channel loss probability is not too high..

EXPERIMENT 2 results



Experiment 2 deals with the variation in throughput with respect to the window sizes. As we have seen earlier, for a lossless channel, the throughput for GBN should be fairly good. Since ABT does not use window size , there will be no change in throughput with respect to window size.

For GBN, we can see that the throughput is 0.02 which is fairly good. It means that almost all the packets were received by the receiver B. This is expected as the window size is

Snippet:

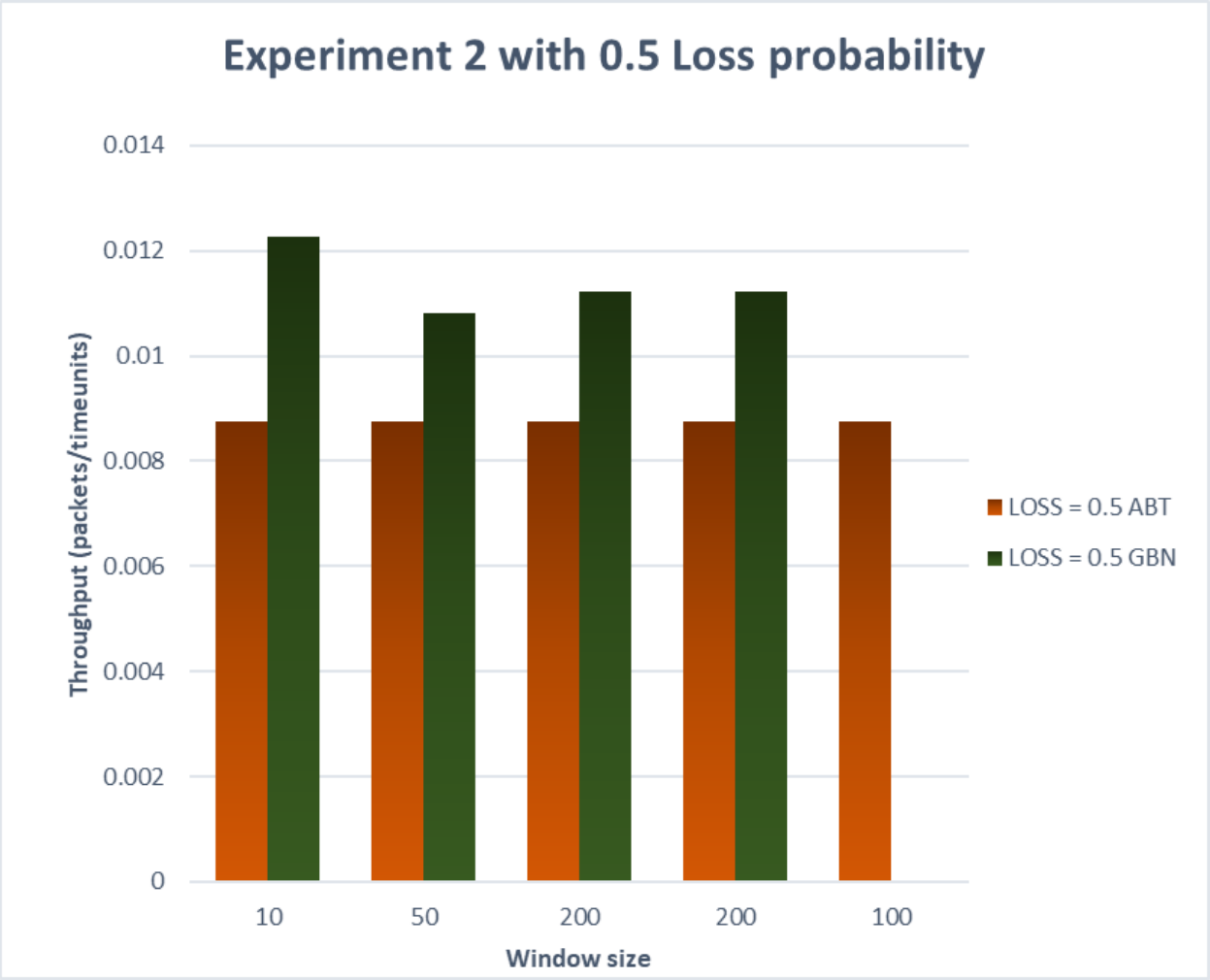
ABT:

```
Transmission time 50019.355469 Simulator terminated at time 50034.355469
 after sending 1000 msgs from layer5

[PA2]1000 packets sent from the Application Layer of Sender A[/PA2]
[PA2]1766 packets sent from the Transport Layer of Sender A[/PA2]
[PA2]1414 packets received at the Transport layer of Receiver B[/PA2]
[PA2]726 packets received at the Application layer of Receiver B[/PA2]
[PA2]Total time: 50034.355469 time units[/PA2]
[PA2]Throughput: 0.014510 packets/time units[/PA2]
```

GBN:

```
Transmission time of oldest unack'd packt 50214.406250
NextSeqNum 1000 Simulator terminated at time 50234.406250
 after sending 1000 msgs from layer5

[PA2]1000 packets sent from the Application Layer of Sender A[/PA2]
[PA2]2413 packets sent from the Transport Layer of Sender A[/PA2]
[PA2]1917 packets received at the Transport layer of Receiver B[/PA2]
[PA2]999 packets received at the Application layer of Receiver B[/PA2]
[PA2]Total time: 50234.406250 time units[/PA2]
[PA2]Throughput: 0.019887 packets/time units[/PA2]
euston {/local/Spring_2023/dvasprab/cse489589_assignment2/dvasprab} > date
Thu May  4 20:11:59 EDT 2023
```

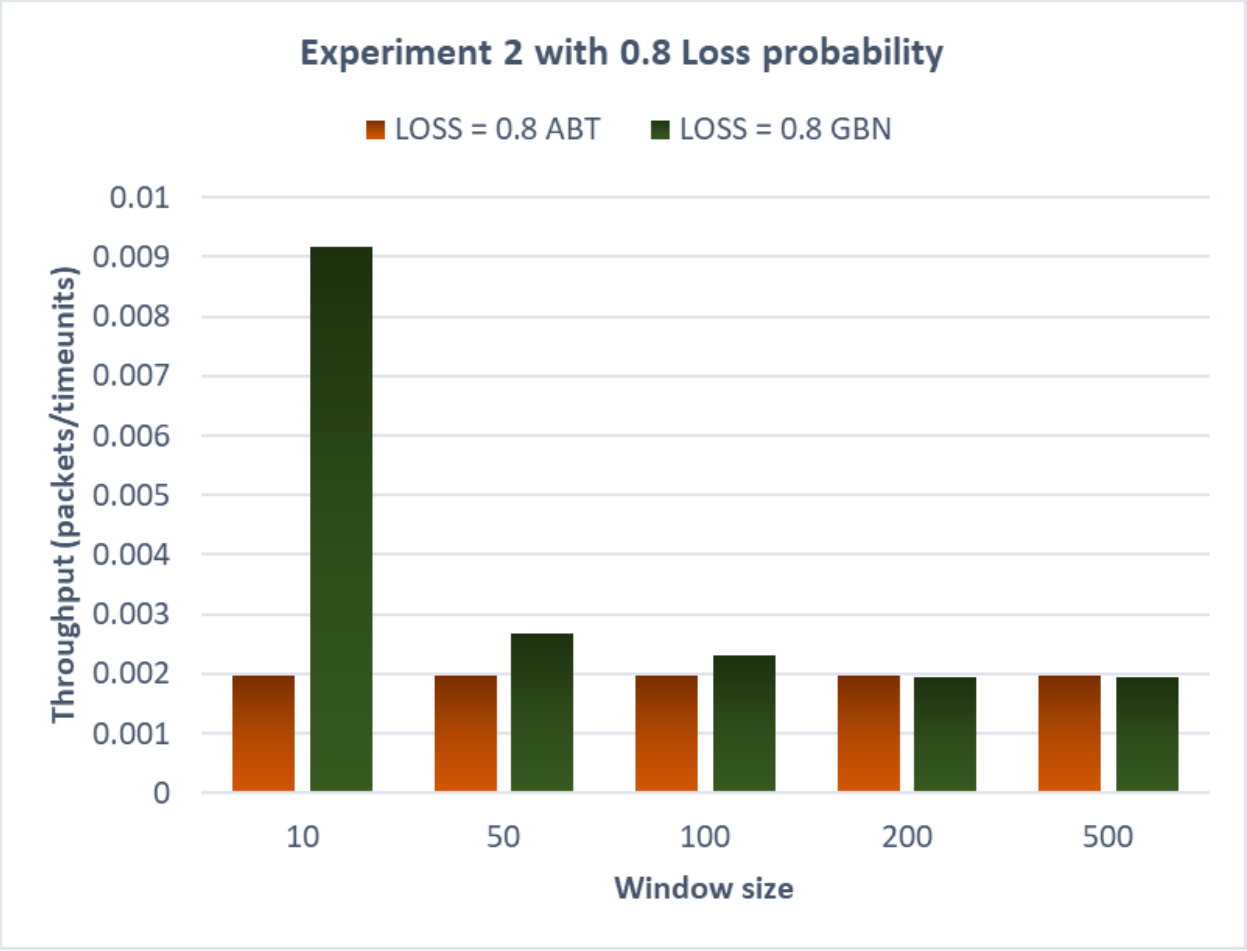
```
    72  20:10   ./gbn -s 1234 -t 50 -m 1000 -w 10 -l 0.2 -c 0.2 -v 1
    75  20:11   ./gbn -s 1234 -t 50 -m 1000 -w 500 -l 0.2 -c 0.2 -v 1
    76  20:11   date
    77  20:12   history
euston {/local/Spring_2023/dvasprab/cse489589_assignment2/dvasprab} >
```



For ABT with 0.5 loss , we can see that the throughput dips to almost half the value : 0.009 than it was when the loss probability was 0.2.

```
[PA2]1000 packets sent from the Application Layer of Sender A[/PA2]
[PA2]2487 packets sent from the Transport Layer of Sender A[/PA2]
[PA2]1248 packets received at the Transport layer of Receiver B[/PA2]
[PA2]433 packets received at the Application layer of Receiver B[/PA2]
[PA2]Total time: 50779.425781 time units[/PA2]
[PA2]Throughput: 0.008527 packets/time units[/PA2]
euston {/local/Spring_2023/dvasprab/cse489589_assignment2/dvasprab} > PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTY
```

The similar trend is observed in the graphs chart of GBN , with few deviations with respect to window size. This depends on the test cases, but averaging the results, we can see that the loss has significantly reduced the throughput from an average of 0.02 to 0.012.



However when it comes to the loss probability being 0.8, we see that the throughput significantly reduces for both ABT and GBN. It is strange to notice that GBN provides 0.009 throughput at window size 10. This is due to the smaller number of retransmits being due to smaller window size 10, and thus comparatively better throughput compared with larger window size that result in high amount of retransmits in lossy channels.

Challenges faced:
High packet loss even with low loss channel and corruption probabilities.

- Changed the A_timerinterrupt logic. Also kept track of the acknowledgement using the **A_ack_received** variable on the sender side.
- Re-transmitted the message packets with ack_number which was checked at B_input(). Discarded the message at receiver B if the message was already received / delayed / or lost. ACK packet for the next message was re-sent again to sender A.
- Re-transmitted ACK packets seq_num checked at A_input(). Discarded if the ACK was already received earlier.

GBN:
GBN baseline implementation worked for lossless and corruption less channel. Fix was required for lossy/corruption inducing channels.

- Tweaked the ACK's implementation on the receiver end . The check on the sender side for each (ACK # == base) where base is the oldest unacknowledged packet's sequence number was causing an issue. Newer message ACK's were being ignored. Change was implemented for the sender to receive any ACK whose sequence number is greater than the base.

```
int ACK_order_status(struct pkt *p, int base)
{
  if (p->acknum >= base)
    return 1;
  else
    return 0;
}
```

(base is the oldest un-acknowledged packet number.[Cumulative ACK's implemented]. Much less re-transmissions were seen after this.

- Handled incorrect order of packets during storage/retrieval. When the nextseqnum is not yet at the end of the window, there is still room for transmitting packets. On calling A_output for the message from the application layer, the next available space is used to store the packet and transmit it. However , if at a previous point in time, a packet was already saved due to the window being full, then this buffer packet should be transmitted first. This is similar to a queue, FIFO principle. This solved the mismatch in expected and received packet.

```
if (nextseqnum < (base + Window_size))
{
  if (buffer_empty(send_buffer[nextseqnum],nextseqnum))
  {
    // If buffer value is empty create packet and put values.
    packetout = create_pkt(&message, nextseqnum , A_ack_counter);
    create_checksum(&packetout);
    send_buffer[nextseqnum] = packetout;
  }
  else
  {
    printf("\n Store IDX value %d",store_idx);
    packetout = create_pkt(&message, store_idx , A_ack_counter);
    create_checksum(&packetout);
    send_buffer[store_idx] = packetout;
    store_idx++;
  }
```

- A_input logic had a bug in that all the packets were being re-transmitted, after an ACK was received. Only the packets arriving after the nextseqnum are to be transmitted, Also the number of packets to be transmitted depends on the latest ACK number received. Used the difference(base - prev_base) to find this value.

```
 for (int idx = 0 ; idx < (base - prev_base) ; idx++)
{
if (!(buffer_empty(send_buffer[nextseqnum+idx],nextseqnum)))
{
printf("\n\n Transmitting next : %f packet no: %d seqnum: %d
",get_sim_time(),nextseqnum+idx,send_buffer[nextseqnum+idx].seqnum);
tolayer3(0, send_buffer[nextseqnum+idx]);
//printf("\n\n ----DEBUG--- %d",&send_buffer[i].seqnum);
printf("\n\n ----DEBUG--- %d",send_buffer[nextseqnum+idx].seqnum);
nextseqnum++;
```

Brief description of the **timeout scheme** you used in your protocol implementation and why you chose that scheme.
Ans:
ABT: The timeout value was chosen to be 15.0 units. This is because the round trip time of the packet is 5 time units as stated in the project description. Since abt requires every packet to be acknowledged before sending the next one, a value of 10.0 was chosen initially. However this became an issue when the channel was input with loss probabilities and corruption (bit errors). To compensate for the time taken for retransmission, a value of 15.0 was chosen as the timeout value.

GBN:The timeout value for GBN was selected to be 5 units higher than that for abt and kept at 20.0. Considering that GBN uses a fixed window to transmit packets in a contiguous time-frame without waiting for acknowledgements, 20.0 seemed to be the optimal value to let most of the packets delivered to the receiver. GBN works on cumulative acknowledgements so setting the timer too large would not be a good idea.

SR:

Brief description (including references to the **corresponding variables and data structures in your code)** of how you implemented multiple software timers in SR using a single hardware timer.

A:

ABT:

Sender side Variables and data structures:

A_seq_counter : Used to keep track of the sequence number of the currently sent packet.

A_ack_counter : Used to keep track of the acknowledgment number of the currently sent packet.

A_ack_received: Used to keep track of whether the acknowledgement was received for the previously sent packet. If value is 0, retransmission is required. If value is 1, then ACK was received.

A_buffer: Buffer used for storing messages incase the previous packet was not acknowledged**.**

A_pckt_copy: A packet copy of the transmitted message packet containing message data. Required for retransmission.

buffer_index: Used to keep track of where to store the message data in the buffer, when the ack is not received for previous packet, but there exists and incoming message from application layer of A.

Receiver side Variables and data structures:

B_seq_counter: Used to keep track of the next expected sequence number of the received packet.

B_ack_counter : Used to keep track of the next expected acknowledgement number of the received packet

GBN:

Sender side Variables and data structures:

base : Used to keep track of the oldest unacknowledged packet sequence number in the window.

prev_base: Since cumulative acks are used, prev_base is used to keep track of the previous base sequence number , so that when a cumulative ack arrives, then the difference between base and previous base , gives the number of packets that can be transmitted next.

nextseqnum: Used as a counter to increment to next packet sequence number to be transmitted. Should always be less than base + Window_size.

Window_size: This is fetched during initialization to get the window size using getwinsize().

store_idx : Inde to be used incase the buffer window is full and there are incoming packets on the sender. I.e. A_output is called but the window is full. So store_idx keeps track of the index to store in the buffer.

send_buffer: The buffer used for storing and transmitting packets. This is an array of packets, where each element is of the type struct pkt.

Receiver side Variables and data structures:

B_ack_counter : Used for the arrival of the next expected packet number and checking the message order at receiver B.

B_packt_copy: Used for sending the ACK packet for the last received message packet in case a duplicate message is received.

Various functions are implemented in the code for ease of understanding with the comments describing their goal.

Some important ones are

```
/*Function to check the ACK order status on the sender A. Cumulative
acknowledgement*/
int ACK_order_status(struct pkt *p, int base)
{
  if (p->acknum >= base)
    return 1;
  else
    return 0;
}
```

```
/*Function to check the message order status on the receiver B*/
int message_order_status(struct pkt *p, int seqnum)
{
  //if (p->seqnum == (seqnum - 1))
  if (p->seqnum == (seqnum - 1))
    return 1;
  else
    return 0;
}
```

References:

- Pointers in C Explained – They're Not as Difficult as You Think (freecodecamp.org)
- Computer Networking A Top-Down Approach, Global Edition, 8th Edition (Kurose, James, Ross, Keith)
- Selective Repeat / Go Back N (tu-berlin.de) Used for understanding various test cases for lost packets and acknowledgements.
- Map one column to x axis second to y axis in excel chart - Super User