CSE 489/589
Programming Assignment 1 Report

Text Chat Application

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

Rupam Patir
UBIT Name: rupampat
Person Number: 50365531

TABLE OF CONTENTS

1. Overview

For checksumming I have created a common function for all protocols.

```
int create_checksum(struct pkt packet);
```

I have created a data structure to store the linked list of messages.

```
struct buffer {
  struct msg message;
  struct buffer * next_message;
}* buffered_messages;
```

For Selective Repeat, the buffer additionally stores the seqnum of the message as well as whether it was ACKed or not. Further description for SR buffer is in section 1.3.

Question: Brief description of the timeout scheme you used in your protocol implementation and why you chose that scheme.

Answer: I used timeout value 50. I chose this number because it was the smallest number at which the simulator ran quickly. I used the same timeout value for all experiments to ensure a constant hyperparameter across the experiments in Section 2. This is however not the ideal scenario.

A timeout value should reflect packet loss and congestion on the link. Choosing a constant timeout value was therefore unhelpful when the simulator simulated different scenarios. **When packet loss occurs, it means that the network is congested and therefore a higher interrupt interval should occur to ensure messages are not being unnecessarily re-sent.**

One of the main bottlenecks in terms of choosing the timeout value was the performance of GBN. For packet loss, I noticed the timeout does not affect the simulator runtime much. However, for packet corruption, the time it takes for the simulator to run is very long (specifically the basic and advanced tests which check large numbers of messages and high corruption values) for timeout values less than 50. This could be because of the following reasons:

1. The interrupt may be happening unnecessarily for packets that may eventually be correctly received i.e. the interrupt interval is too low for congested scenarios.

2. The effects of a bad timeout interval were heightened with GBN because ALL the messages in the window size are re-sent.

3. Every time an interrupt happens with GBN all the messages in that window are sent again. This means that the number of messages B receives is very high especially for large window sizes. This subsequently means the number of times B processes an incoming message is very high for GBN, slowing down the simulator as it needs to run this function many times.

4. For packet loss, only the interrupt function is run and all the messages in the window are sent. However, for packet corruption and assuming no packet loss, the number of messages bloats up even more because not only does B need to process all the repeat messages from part 1, but A also needs to process all the acknowledgements from host B that may be corrupted. This means the number of functions that are run by the simulator increases even more.

I noticed that for the high timeout value above my code was resistant to these congested simulations. It may be the case that in these congested simulations, unnecessary repeat messages and ACKs were much less and therefore my code runs faster. As such the main factor behind me choosing the timeout value was the speed at which the experiments for congested scenarios were run. The faster the experiments ran meant fewer unnecessary repeat messages were sent.

Because of these reasons, the timeout value I chose was mostly related to the least timeout that resulted in the simulator running *quickly* for congested scenarios with GBN.

Theoretically, however, the timeout interval is determined depending on a correctly received sample's

RTT. A successfully received packet will have it's RTT registered to estimate the timeout interval using the formulae,

$$EstimatedRTT = (1 - \alpha) \cdot EstimatedRTT + \alpha \cdot SampleRTT$$

$$DevRTT = (1 - \beta) \cdot DevRTT + \beta \cdot | SampleRTT - EstimatedRTT |$$

$$TimeoutInterval = EstimatedRTT + 4 \cdot DevRTT$$

This could have been coded into this project by using the `get_sim_time()` function to keep a track of when a packet was sent and when its corresponding acknowledgement was received. However, this was out of the scope of this project and hence was not implemented.

1.1. Alternating Bit Protocol

I created an enum that has two states.

```
enum host_states {
  waiting_for_acknowledgment,
  available
};
```

We have two other global variables,

```
int seqnum_A;
int acknum_B;
```

The alternating bit protocol depends on the host state and presence of messages in the buffer. If a new message is added, it first gets added to the buffer. If the host state is currently waiting for acknowledgement, then nothing happens otherwise the host immediately sends that message to host B. The seqnum of the sent message will be whatever the value of seqnum_A is.

Once the host B receives a message, it carries out the checksum matching and checks if the message received is of the correct seqnum and if the seqnum is the same as the seqnum of acknum_B. If it is, the message is sent to Layer 5, the acknowledgement for the packet is sent back to A, then the acknum_B bit is flipped. If the message received is corrupted or out of order, then the ack for the last correctly received bit is re-sent.

When host B receives an acknowledgement, it checks for corruption and if the ACK is equal to seqnum_A which is the seqnum of the last message. If it is, then the message is removed from the buffer; the seqnum_A field is then flipped; then if there are no messages in the buffer, the state of the host is made available, otherwise the next message in the buffer is sent with sequence number equal to seqnum_A. If the ACK was wrong or corrupted, nothing happens.

When an interrupt happens in A, it resends the first message in the buffer with seqnum_A.

1.2. Go back N Protocol

We have the following global variables;

```
int seqnum_A;
int acknum_B;
int WINDOWSIZE;
int base;
int nextseqnum;
```

For GBN, the process is similar to ABT except now we can send many more messages than ABT. Specifically, we can send a WINDOWSIZE amount of messages.

When A receives a message from Layer 5 to be sent to B, it is assigned the seqnum defined by nextseqnum (which is subsequently incremented), the message is buffered, and then host A checks if that message is within the window of messages able to be sent. This window is defined by base+WINDOSIZE. The base is the last successful message that was ACKed in order. If the seqnum is within this window, the message is immediately sent. The timer is started if the seqnum is the same as the base because this means that the timer was not running (the timer is only started when the current base is sent).

When B receives a message it checks if the received message is corrupted and in order. It checks the latter by using acknum_B which unlike the ABT protocol stores a number (instead of a 0/1 bit) to indicate the expected seqnum from B. If the message received is the current seqnum indicated by acknum_B, then the message is sent to Layer 5 of B and an ACK for it is sent. Subsequently, acknum_B is incremented. If the message was out of order, Host B simply resends the ACK for the last correctly received message which conveniently corresponds to acknum_B - 1;

Now, when A receives an acknowledgement from B, it checks for corruption; If corrupted, nothing happens. If the ACK is greater than the base then all messages upto that ACK were correctly received and these messages are removed from the buffer. It updates the base.Then it stops the timer if there are no more messages to be sent, else it restarts the timer.

When an interrupt happens in A, it resends ALL the messages in the window size from the base.

1.3. Selective Repeat

I have created a different data structure here to store the linked list of messages in the buffer. As can be seen there are two additional fields seqnum and acked.

```
struct buffer {
  struct msg message;
  int seqnum;
  int acked;
  struct buffer *next_message;
};
```

Selective Repeat is similar to Go Back N. However, in Selective Repeat both host A and host B will have a buffer of messages. A message successfully received by B need not be sent again and if it was out of order, it'll be stored in B's buffer.

When A receives a message from Layer 5 to be sent to B, it is assigned the seqnum defined by nextseqnum (which is consecutively incremented), the message is buffered, and then host A checks if that message is within the window of messages able to be sent. This window is defined by the base+WINDOSIZE. The base is the last successful message that was ACKed in order. If the seqnum is within this window, the message is immediately sent. This is the same as in GBN. However, the timer functionality here is different. There is a unique timer for each message that is sent by A.

Question: 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.

Answer: For the SR Timer I created a data structure called timer.

```
struct timer {
  int seqnum;
  double absolute_interrupt_time;
  struct timer *next_timer;
};
```

The timer is essentially a linked list that stores the seqnum of the current timer and the time at which that timer should cause an interrupt. Since it's a linked list, it also stores the next timer in the list.

What happens here is that when a new interrupt needs to be added we add it to the end of the list, with that relevant interrupt time. Since all the timers are of equal length we need not worry about where we have to put this timer. It will always be at the end of the list since all other timers before it would have an interrupt time before it. Now we set the hardware timer to the head of this linked list. When that interrupts, we remove that timer from the linked list if that message was already acknowledged. We can check if it was acknowledged by checking the list of buffered messages for that seqnum. The buffer contains a field called ack that stores this information. If the seqnum of the timer was not found in the buffer, this also means that the message was ACKed. If it was not acknowledged, we remove it from the head of the list and append it again at the end of the list with the new interrupt time.

In this way, every time a new message is added, a new timer is maintained for it. The `starttimer()` function just sets the timer to whatever the absolute_interrupt_time is at the head of this linked list.

For example, for a timeout interval of 50 and the following series of events

1. At T=0, Message A sent with Seq 1 and also added to buffer;
2. At T =10, Message A is acknowledged and the buffer reflects this;
3. At T= 20, Message B sent with Seq 2 and also added to buffer;
4. At T = 60, Message C sent with Seq 3 and also added to buffer;
5. At T = 80, Message B is acknowledged and the buffer reflects this;
6. At T = 90, Message C is acknowledged and the buffer reflects this;

Now,

At T=0; Add new timer; starttimer(50);

Message A (HEAD)
Seq No. 1
Interrupt TIme 50

At T=20; Add new timer;

Message A (HEAD)		**Message B**
Seq No. 1	------->	Seq No. 2
Interrupt TIme 50		Interrupt TIme 70

At T=50 (Interrupt!); Remove head since A is Acknowledged; starttimer(70);

Message B (HEAD)
Seq No. 2
Interrupt TIme 70

At T=60; Add new timer;

Message B (HEAD)		**Message C**
Seq No. 2	------->	Seq No. 3
Interrupt TIme 70		Interrupt TIme 110

At T=70 (Interrupt!); Message B not yet ACKed. Resend it; Remove timer head and add it again to the end of the list; starttimer(70);

Message C (HEAD)		**Message B**
Seq No. 3	------->	Seq No. 2
Interrupt TIme 110		Interrupt TIme 120

At T=110 (Interrupt!); Remove timer since Message C is ACKed

Message B (HEAD)
Seq No. 2
Interrupt TIme 120

At T=120 (Interrupt!); Remove timer since Message C is ACKed

Linked List EMPTY

When B receives a message from A, it checks for corruption. If it's corrupted, nothing happens. If it's not corrupted, it checks if the seqnum is the next expected seqnum. If it is less than the expected seqnum, that means that message was already successfully sent to B's layer 5. If it is greater than the expected seqnum, the message is buffered on B's buffer and an ACK for it is sent to A. If it is equal to the expected seqnum, then it as well as all the consecutive messages (in order messages) in the buffer are sent to B's Layer 5. These messages are also removed from B's buffer. An ACK for that seqnum is sent from B to A.

When A receives an ACK from B, it checks if it's corrupted or not. If not corrupted, it sets the relevant message in the buffer to acknowledge i.e. the acked field is set to True. Then A checks the head of it's buffer and removes it and all subsequent messages if they are all ACKed. The global timer is also updated such that their

timers are removed from the global timer linked list. The hardware timer is reset to timeout at the new head of the global timer linked list. If the message ACKed from B was not the base i.e. it was not the head of the buffer, nothing happens (except flipping the relevant message's acked field to True as mentioned earlier).

When A interrupts, A checks if that message was ACKed in the buffer. If it wasn't, then the timer is removed from the head of the global timer and then it is readded to the end of the list with the new interrupt time. If it was ACKed, then the timer is simply removed from the head of the global timer. Now, the timer is started as indicated by the new head of the global timer. If there are no more timers in the global timer, then nothing happens.

2. EXPERIMENTAL OBSERVATIONS

In this section we present the experimental observations. Specifically, we compare throughput vs loss probability as well as throughput vs window size. In each of the following 2 experiments, we ran each of the 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. We use timeout 50 for all the experiments.

NOTE: I have not presented the output of the run_experiments script in this report because it is too large to be viewed on this report pdf. However, it can be accessed on Google Sheets where it is presented clearly. Click **here**[1] (full link in text given in the footer at the end of this page in case it is not clickable) to view the complete experimental results on google sheets where it is much clearer and cleanly presented.

2.1. Throughput vs Loss Probability

With loss probabilities: {0.1, 0.2, 0.4, 0.6, 0.8}, we compare the 3 protocols' throughputs at the application layer of receiver B. We use 2 window sizes: {10, 50} for the Go-Back-N version and the Selective-Repeat Version.

From Figure 1, 2 and 3, we see that in all cases the throughput decreases as we increase the loss probability. However, the throughput for GBN and SR are higher than ABT. This is so because more messages are sent into the link in the case of GBN and SR than with ABT.

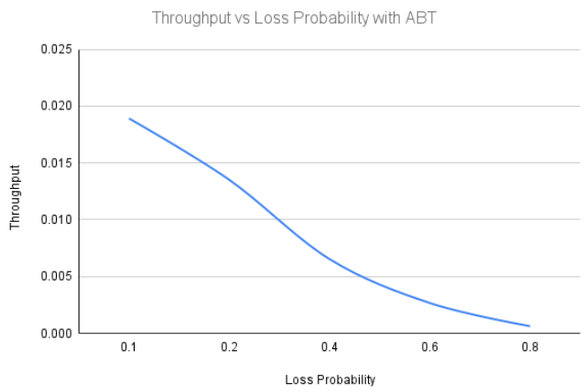


Figure 1: Throughput vs Loss Probability with ABT

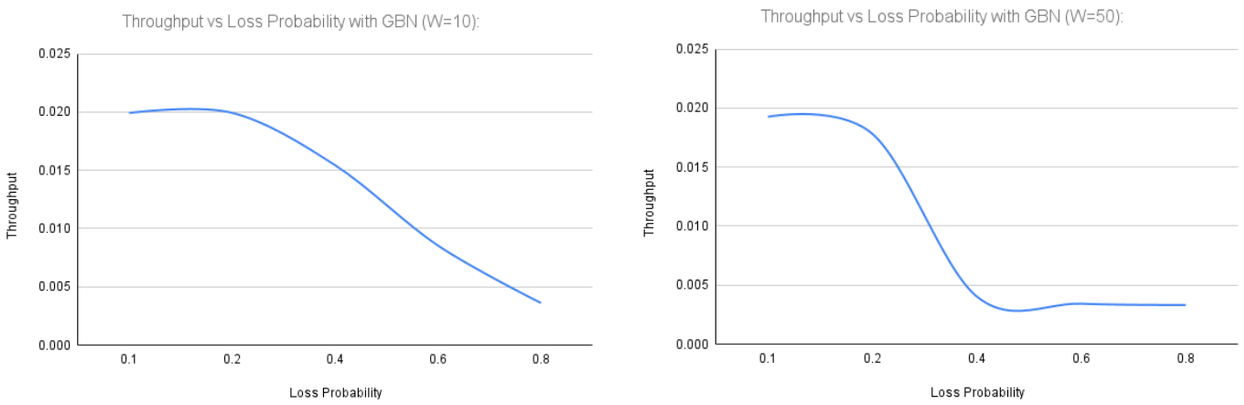


Figure 2: Throughput vs Loss Probability with GBN; and Window Size 10 (left) and 50 (right)

[1] https://docs.google.com/spreadsheets/d/e/2PACX-1vRopO9wHI7y-YgK0fGf60ndhc-dFjI9s8mmNpwltISDgHRQe6hKEyDLZfjZO7uqBgA6F0ZBiJ9iInN7/pubhtml

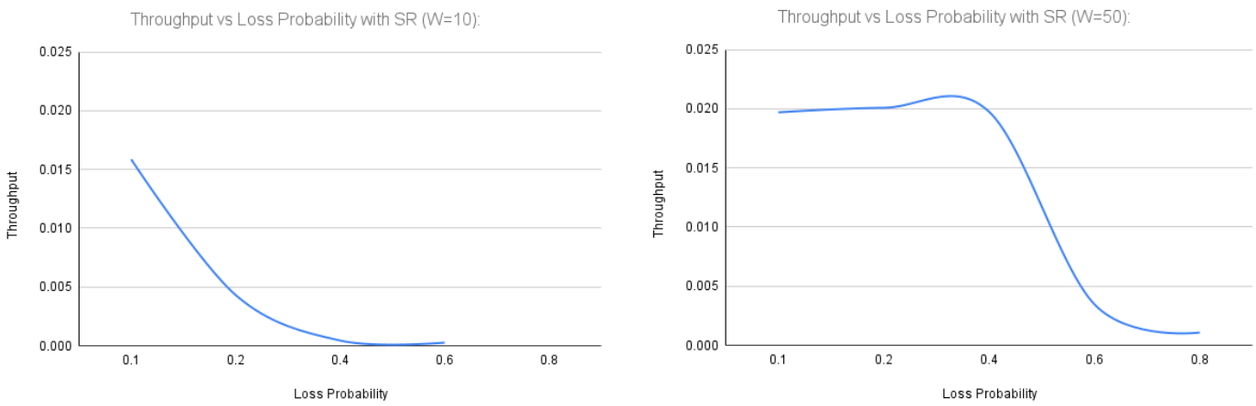


Figure 3: Throughput vs Loss Probability with SR; and Window Size 10 (left) and 50 (right)

Between GBN and SR, with window size of 10 we see that the throughput of GBN is generally higher than that of SR. Unlike GBN, SR does not send unnecessary repeat messages that are already received by B into the link. As such, SR has a smaller throughput than GBN. With a window size of 50, however, SR has a higher throughput. This could be because SR is better able to utilise the links bandwidth under congested scenarios and tolerate a higher loss probability.

2.2. Throughput vs Window Size

With window sizes: {10, 50, 100, 200, 500} for GBN and SR, we compare the 3 protocols' throughputs at the application layer of receiver B. We use 3 loss probabilities: {0.2, 0.5, 0.8} for all 3 protocols.

Figure 4: Throughput vs Window Size with GBN for Loss Probabilities {0.2, 0.5, 0.8}

From Figure 4 we see that the GBN protocol's throughput first decreases and then becomes constant (after a window size of 50-100) as we increase the window size. As we increase the window size, the link becomes more and more congested and throughput becomes lesser. As we increase the loss probability the throughput becomes much lesser as well. This is because as loss probability is increased, the number of messages received by B will be much lesser.

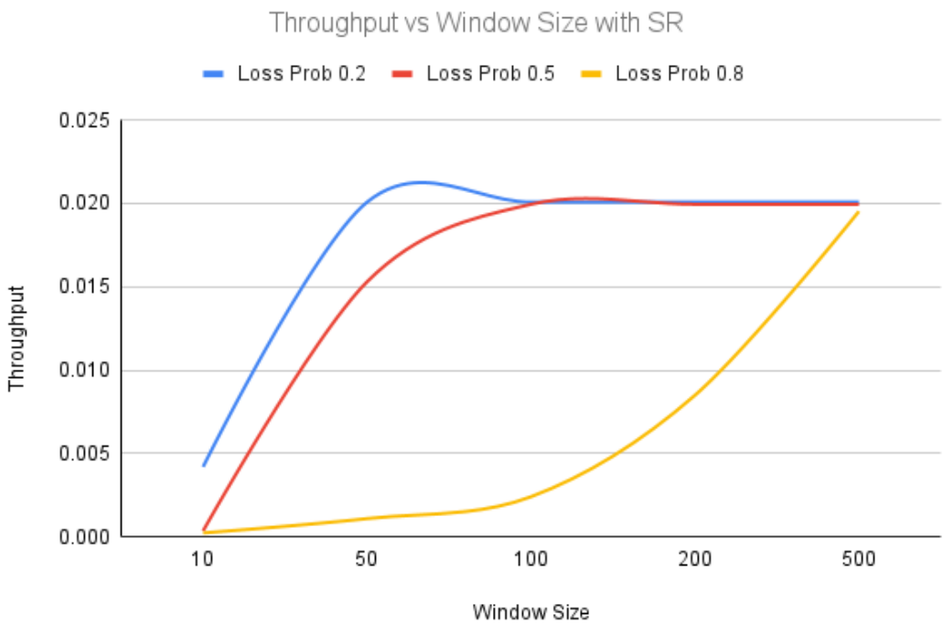


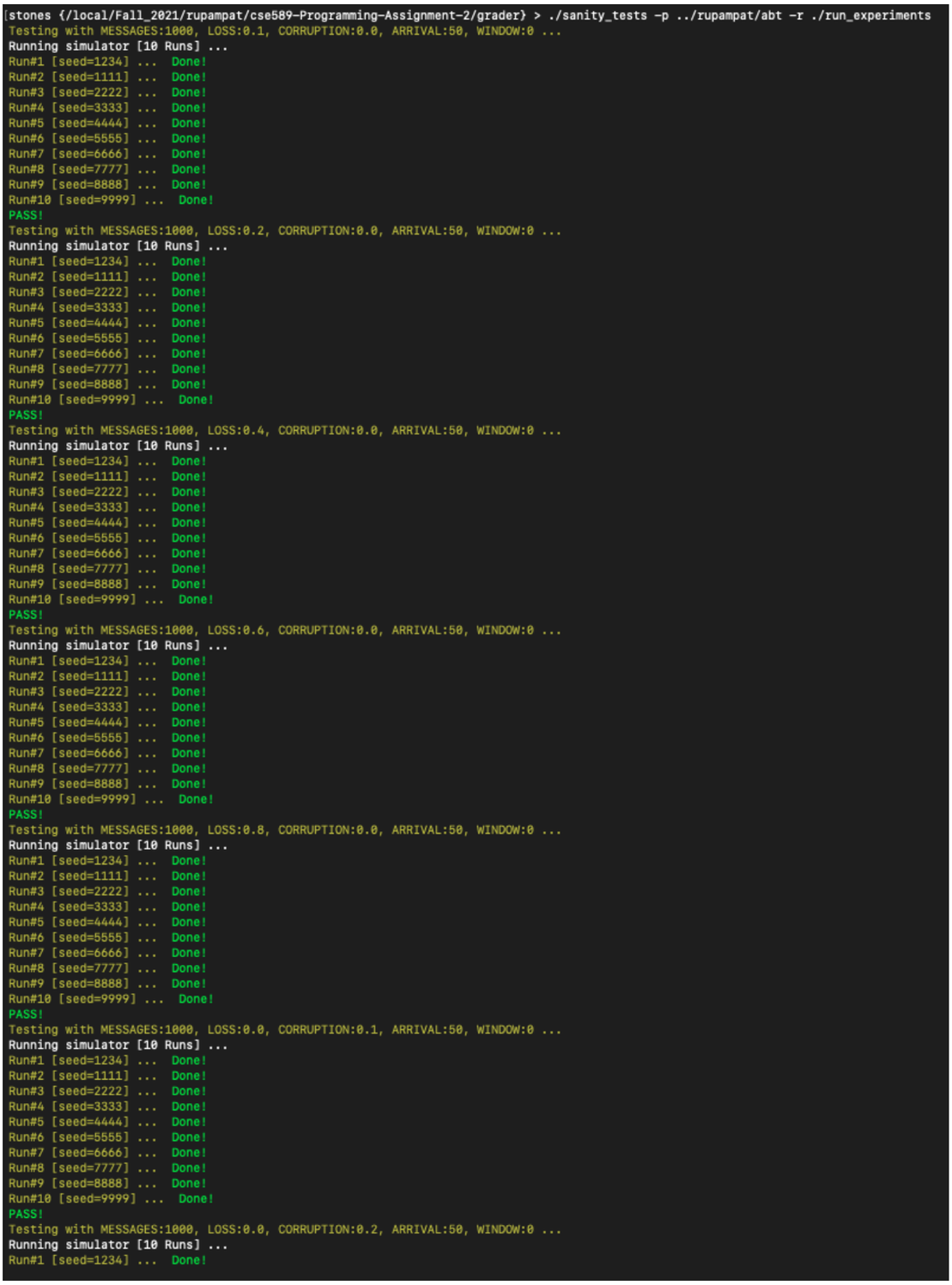
Figure 5: Throughput vs Window Size with SR for Loss Probabilities {0.2, 0.5, 0.8}

From Figure 5 we see that the SR protocol's throughput increases logarithmically as we increase the window size. We see that the SR protocol utilises the bandwidth of the link the best by sending as many messages into the link as possible and preventing unnecessary repeat messages. We see that as we increase the loss probability the throughput decreases. This is because the chance of a packet being received is lesser and therefore throughput will be lesser as well. We see that the SR is much better at tolerating high loss probabilities than GBN is.

3. SCREENSHOTS

3.1. SANITY TESTS

3.1.1. Alternating Bit Protocol



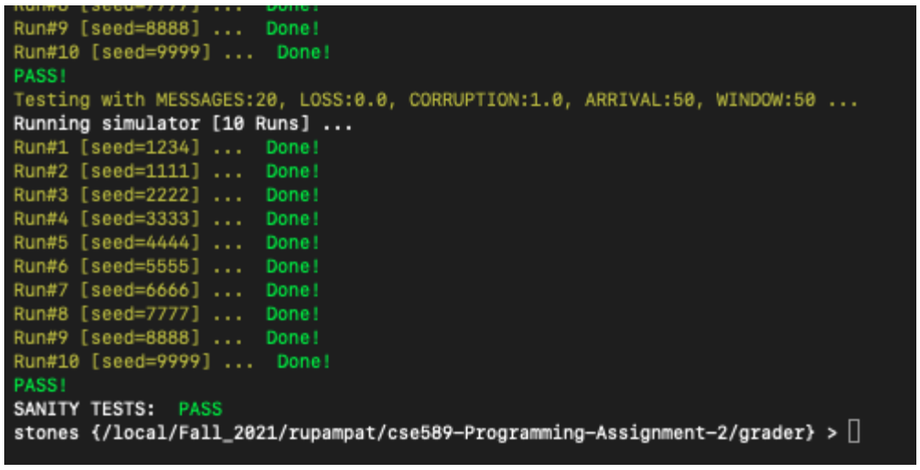
```
Run#9 [seed=8888] ...  Done!
Run#10 [seed=9999] ...  Done!
PASS!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.2, ARRIVAL:50, WINDOW:0 ...
Running simulator [10 Runs] ...
Run#1 [seed=1234] ...  Done!
Run#2 [seed=1111] ...  Done!
Run#3 [seed=2222] ...  Done!
Run#4 [seed=3333] ...  Done!
Run#5 [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 ...
```

```
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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
```

3.1.2. Go Back N

```
[stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./sanity_tests -p ../rupampat/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!
```

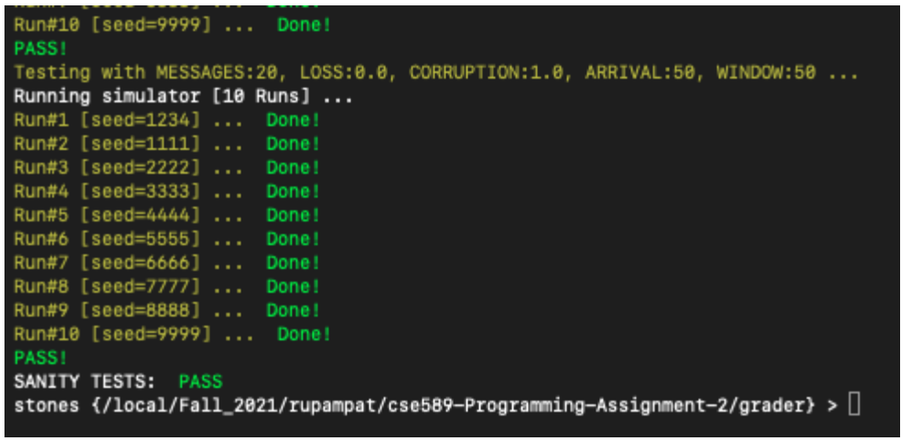
```
Run#10 [seed=9999] ...  Done!
PASS!
Testing with MESSAGES:1000, LOSS:0.0, CORRUPTION:0.2, ARRIVAL:50, WINDOW:10 ...
Running simulator [10 Runs] ...
Run#1 [seed=1234] ...  Done!
Run#2 [seed=1111] ...  Done!
Run#3 [seed=2222] ...  Done!
Run#4 [seed=3333] ...  Done!
Run#5 [seed=4444] ...  Done!
Run#6 [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#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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >

3.1.3. Selective Repeat

```
[stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./sanity_tests -p ../rupampat/sr -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#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#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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >

3.2. BASIC TESTS

3.2.1. Alternating Bit Protocol

```
[stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./basic_tests -p ../rupampat/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
stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
```

3.2.2. Go Back N

```
[stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./basic_tests -p ../rupampat/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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
```

3.2.3. Selective Repeat

```
[stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./basic_tests -p ../rupampat/sr -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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
```

3.3. ADVANCED TESTS

3.3.1. Alternating Bit Protocol

```
(stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./advanced_tests -p ../rupampat/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!
```

```
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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
```

3.3.2. Go back N

```
[stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./advanced_tests -p ../rupampat/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#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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
```

3.3.3. Selective Repeat

```
[stones {/local/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} > ./advanced_tests -p ../rupampat/sr -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#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/Fall_2021/rupampat/cse589-Programming-Assignment-2/grader} >
```