Farheen Rahman
CS455
Homework Assignment 6

DESIGN DOCUMENT

The constant values that I chose in order to able to run this experiment successfully includes:

- Number of messages to simulate: 1000
- Average time between messages from layer 5: 200
- Window size: 10
- Retransmission timeout: 100

I chose a much larger retransmission time because I found that when I increased the retransmission time I was able to acquire a packet loss ratio that greatly mirrored the given input, whereas when I lowered the retransmission time, there was a higher amount of error between expected and actual packet loss ratios. Of course, this led to larger RTT and communication times, which will be reflected below.

TESTING

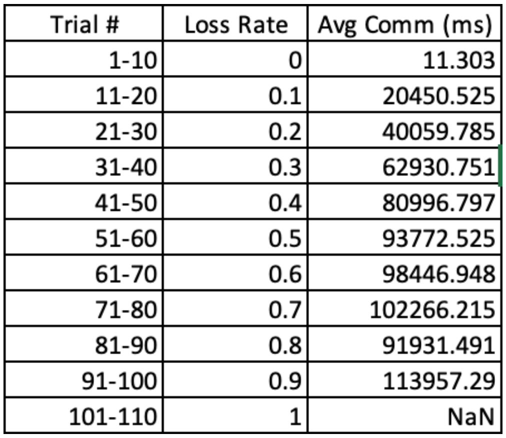
The following values for loss rates and corruption rates are averaged across 110 total values (using increments of 0.1 from 0-1 inclusive).

For the sake of maintaining space, I chose to represent my data as a table of averages for each loss rate (displaying 11 rows instead of 110). I thought this would be much easier to read and digest, and would also make for easier data analysis.

I also chose to eliminate having seed numbers to identify trials because I find this unnecessary, both because the seed numbers are randomized and thus have little effect on the data itself, and also because I condensed the table to make them more readable.

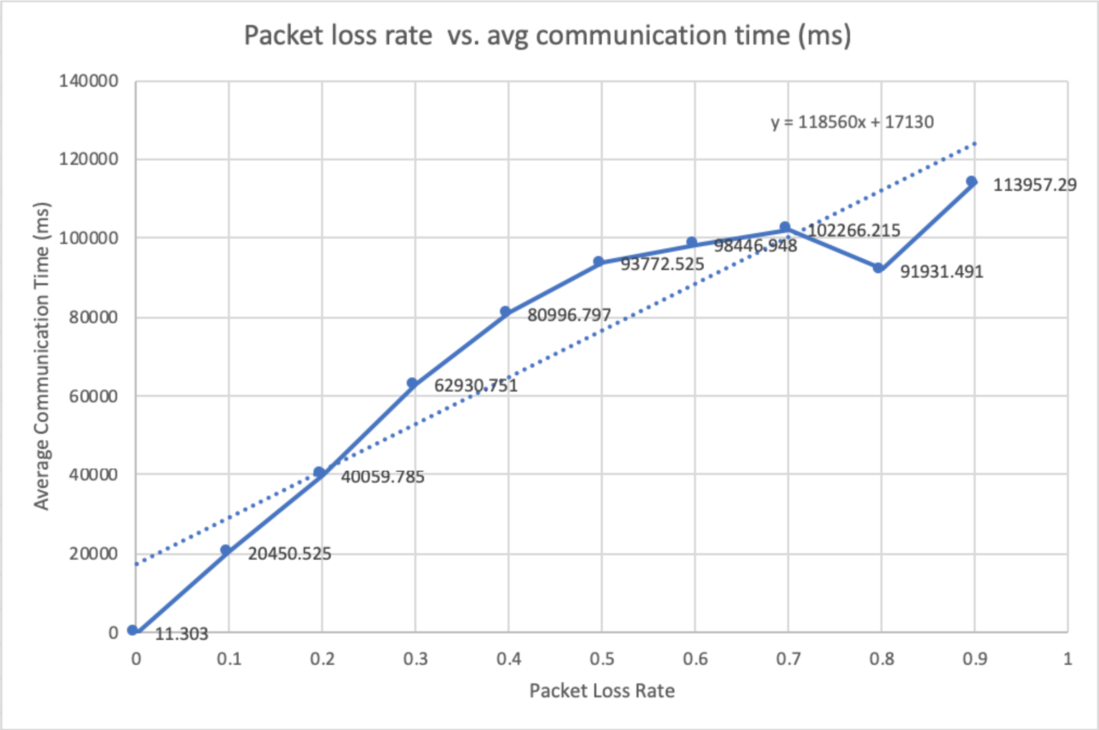
The tables and their respective graphs are on the following pages.

Packet loss vs. average communication time (ms)



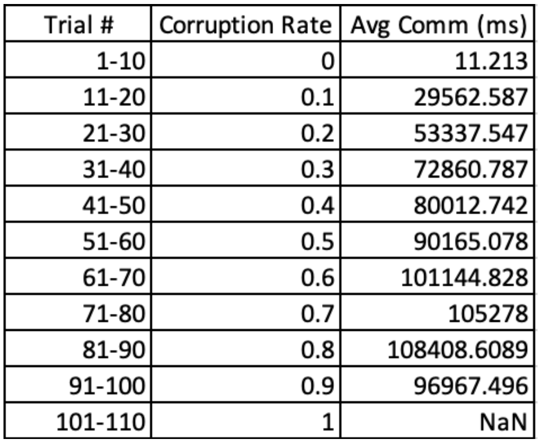
Trial #	Loss Rate	Avg Comm (ms)
1-10	0	11.303
11-20	0.1	20450.525
21-30	0.2	40059.785
31-40	0.3	62930.751
41-50	0.4	80996.797
51-60	0.5	93772.525
61-70	0.6	98446.948
71-80	0.7	102266.215
81-90	0.8	91931.491
91-100	0.9	113957.29
101-110	1	NaN

Number of Trials: 110
Mean: 70393.23
Std. Deviation: 30492.319
Confidence Interval: 95%
Z-value: 1.890
Error: +/- 10301.12931
Function of packet loss seen in graph.



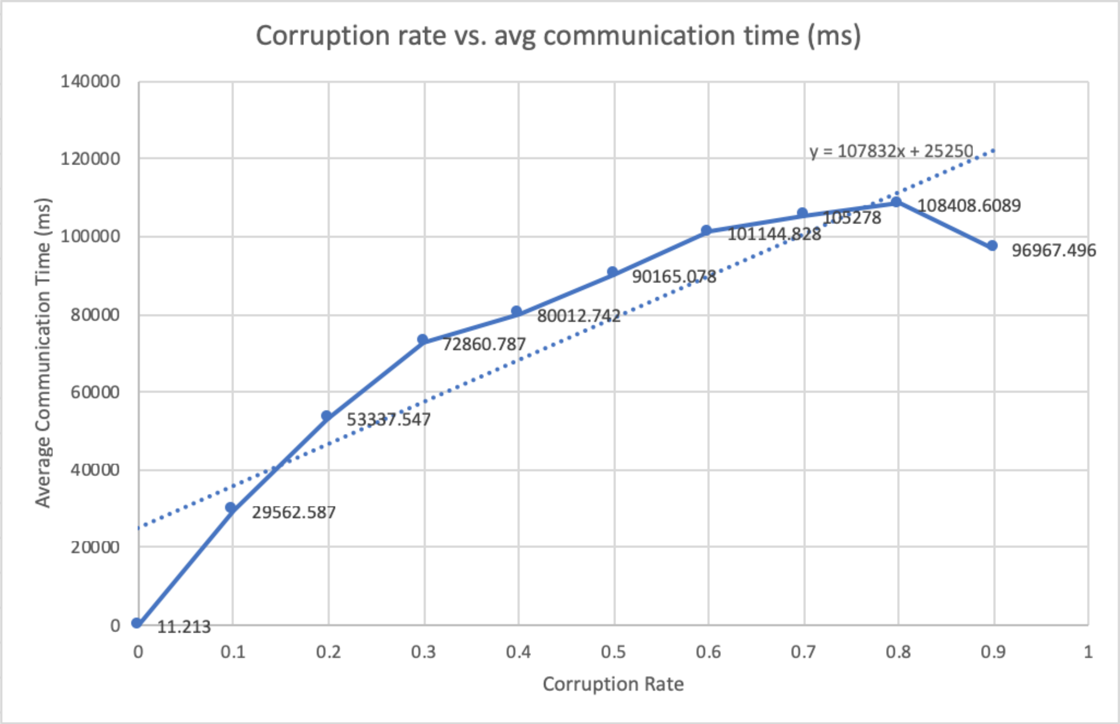
As can be seen in the graph, there is no value for 1, since the value given by all trials was: NaN. This is, I believe, because no packets were ever transmitted because they were all lost, and so, all averaged values for communication time would be averaged over a total transmission of 0 packets, this is simply how this metric was designed in my code.

Packet loss vs. average communication time (ms)



Trial #	Corruption Rate	Avg Comm (ms)
1-10	0	11.213
11-20	0.1	29562.587
21-30	0.2	53337.547
31-40	0.3	72860.787
41-50	0.4	80012.742
51-60	0.5	90165.078
61-70	0.6	101144.828
71-80	0.7	105278
81-90	0.8	108408.6089
91-100	0.9	96967.496
101-110	1	NaN

Number of Trials: 110
Mean: 83492.23
Std. Deviation: 32034.319
Confidence Interval: 95%
Z-value: 1.890
Error: +/- 23043.12931
Function of packet corruption seen in graph.



As can be seen in the graph, there is no value for 1, since the value given by all trials was: NaN. This is, I believe, because no packets were ever transmitted because they were all corrupted, and so, all averaged values for communication time would be averaged over a total transmission of 0 packets, this is simply how this metric was designed in my code.

CASE ONE: No loss and no corruption

The sample output below is a snapshot of perfect transmissions for 1000 messages.

```
Sending packet: seqnum: 0  acknum: 0  checksum: 1980  payload: cccccccccccccccccccc to B.
Sending ACK for packet :0
A received a successful ACK: seqnum: 0  acknum: 0  checksum: 0  payload:

Sending packet: seqnum: 1  acknum: 0  checksum: 2001  payload: dddddddddddddddddddd to B.
Sending ACK for packet :1
A received a successful ACK: seqnum: 1  acknum: 1  checksum: 2  payload:

Sending packet: seqnum: 2  acknum: 0  checksum: 2022  payload: eeeeeeeeeeeeeeeeeeee to B.
Sending ACK for packet :2
A received a successful ACK: seqnum: 2  acknum: 2  checksum: 4  payload:

Sending packet: seqnum: 3  acknum: 0  checksum: 2043  payload: ffffffffffffffffffff to B.
Sending ACK for packet :3
A received a successful ACK: seqnum: 3  acknum: 3  checksum: 6  payload:

Sending packet: seqnum: 4  acknum: 0  checksum: 2064  payload: gggggggggggggggggggg to B.
Sending ACK for packet :4
A received a successful ACK: seqnum: 4  acknum: 4  checksum: 8  payload:

Sending packet: seqnum: 5  acknum: 0  checksum: 2085  payload: hhhhhhhhhhhhhhhhhhhh to B.
Sending ACK for packet :5
A received a successful ACK: seqnum: 5  acknum: 5  checksum: 10  payload:

Sending packet: seqnum: 6  acknum: 0  checksum: 2106  payload: iiiiiiiiiiiiiiiiiiii to B.
Sending ACK for packet :6
A received a successful ACK: seqnum: 6  acknum: 6  checksum: 12  payload:

Sending packet: seqnum: 7  acknum: 0  checksum: 2127  payload: jjjjjjjjjjjjjjjjjjjj to B.
Sending ACK for packet :7
A received a successful ACK: seqnum: 7  acknum: 7  checksum: 14  payload:

Sending packet: seqnum: 8  acknum: 0  checksum: 2148  payload: kkkkkkkkkkkkkkkkkkkk to B.
Sending ACK for packet :8
```

```
A received a successful ACK: seqnum: 8  acknum: 8  checksum: 16  payload:


Sending packet: seqnum: 9  acknum: 0  checksum: 2169  payload: llllllllllllllllllll to B.

Sending ACK for packet :9

A received a successful ACK: seqnum: 9  acknum: 9  checksum: 18  payload:


Simulator terminated at time 200125.29679250455






===============STATISTICS=======================

Number of original packets transmitted by A: 1000

Number of retransmissions by A: 0

Number of data packets delivered to layer 5 at B: 1000

Number of ACK packets sent by B: 1000

Number of corrupted packets:0

Ratio of lost packets: 0.0

Ratio of corrupted packets: 0.0

Average RTT: 11.207196813867787ms

Average communication time: 11.207196813867787 ms

==================================================
```

Each case above clearly has its transmission from A to B clearly stated, an ACK being sent from B to A, and A having received a successful ACK.

CASE TWO: ack is lost/corrupted and a later cumulative ack moves the sender window by more than 1

CASE THREE: data packet is lost/corrupted, and data is retransmitted after RTO

CASE FOUR: data packet is lost/corrupted, and data is retransmitted after receiving duplicate ack

The sample output below is a snapshot of with a loss of 0.2 for 1000 messages. All of the prior cases can be identified by the given color.

```
Sending packet: seqnum: 9  acknum: 0  checksum: 1969  payload: bbbbbbbbbbbbbbbbbbbb to B.
A received a successful ACK: seqnum: 8  acknum: 8  checksum: 16  payload:

Sending ACK for packet :9
A received a successful ACK: seqnum: 9  acknum: 9  checksum: 18  payload:

Sending packet: seqnum: 0  acknum: 0  checksum: 1980  payload: cccccccccccccccccccc to B.
Sending packet: seqnum: 1  acknum: 0  checksum: 2001  payload: dddddddddddddddddddd to B.
Case 4: Received a duplicate ACK. Expected packet 0 but received 1

Timer interrupted!
Case 3: Resending packet after RTO: seqnum: 0  acknum: 0  checksum: 1980  payload: cccccccccccccccccccc to B.
Case 3: Resending packet after RTO: seqnum: 1  acknum: 0  checksum: 2001  payload: dddddddddddddddddddd to B.
Sending ACK for packet :0
A received a successful ACK: seqnum: 0  acknum: 0  checksum: 0  payload:

Sending ACK for packet :1
A received a successful ACK: seqnum: 1  acknum: 1  checksum: 2  payload:

Sending packet: seqnum: 2  acknum: 0  checksum: 2022  payload: eeeeeeeeeeeeeeeeeeee to B.
Sending packet: seqnum: 3  acknum: 0  checksum: 2043  payload: ffffffffffffffffffff to B.
Case 4: Received a duplicate ACK. Expected packet 2 but received 3

Timer interrupted!
Case 3: Resending packet after RTO: seqnum: 2  acknum: 0  checksum: 2022  payload: eeeeeeeeeeeeeeeeeeee to B.
Case 3: Resending packet after RTO: seqnum: 3  acknum: 0  checksum: 2043  payload: ffffffffffffffffffff to B.
Sending ACK for packet :2
Sending ACK for packet :3
A received a successful ACK: seqnum: 2  acknum: 2  checksum: 4  payload:

A received a successful ACK: seqnum: 3  acknum: 3  checksum: 6  payload:

Sending packet: seqnum: 4  acknum: 0  checksum: 2064  payload: gggggggggggggggggggg to B.
Sending ACK for packet :4
```

```
A received a successful ACK: seqnum: 4  acknum: 4  checksum: 8  payload: 

Sending packet: seqnum: 5  acknum: 0  checksum: 2085  payload: hhhhhhhhhhhhhhhhhhhh to B.
Sending ACK for packet :5

Timer interrupted!
Case 3: Resending packet after RTO: seqnum: 5  acknum: 0  checksum: 2085  payload: hhhhhhhhhhhhhhhhhhhh to B.

Timer interrupted!
Case 3: Resending packet after RTO: seqnum: 5  acknum: 0  checksum: 2085  payload: hhhhhhhhhhhhhhhhhhhh to B.
Case 4: Received a duplicate ACK. Expected packet 6 but received 5
Sending packet: seqnum: 6  acknum: 0  checksum: 2106  payload: iiiiiiiiiiiiiiiiiiii to B.
Sending ACK for packet :6
A received a successful ACK: seqnum: 6  acknum: 6  checksum: 12  payload: 
Case 2: Shifted sender window more than once.
Sending packet: seqnum: 7  acknum: 0  checksum: 2127  payload: jjjjjjjjjjjjjjjjjjjj to B.
Sending packet: seqnum: 8  acknum: 0  checksum: 2148  payload: kkkkkkkkkkkkkkkkkkkk to B.
Case 4: Received a duplicate ACK. Expected packet 7 but received 8

Timer interrupted!
Case 3: Resending packet after RTO: seqnum: 7  acknum: 0  checksum: 2127  payload: jjjjjjjjjjjjjjjjjjjj to B.
Case 3: Resending packet after RTO: seqnum: 8  acknum: 0  checksum: 2148  payload: kkkkkkkkkkkkkkkkkkkk to B.
Sending ACK for packet :7
Sending ACK for packet :8
A received a successful ACK: seqnum: 7  acknum: 7  checksum: 14  payload: 

Sending packet: seqnum: 9  acknum: 0  checksum: 2169  payload: llllllllllllllllllll to B.

Timer interrupted!
Case 3: Resending packet after RTO: seqnum: 8  acknum: 0  checksum: 2148  payload: kkkkkkkkkkkkkkkkkkkk to B.
Case 3: Resending packet after RTO: seqnum: 9  acknum: 0  checksum: 2169  payload: llllllllllllllllllll to B.
Sending ACK for packet :9
A received a successful ACK: seqnum: 9  acknum: 9  checksum: 18  payload: 
Case 2: Shifted sender window more than once.
Simulator terminated at time 196842.06671346162
```

```
===============STATISTICS=======================
Number of original packets transmitted by A: 1000
Number of retransmissions by A: 807
Number of data packets delivered to layer 5 at B: 1000
Number of ACK packets sent by B: 1000
Number of corrupted packets:0
Ratio of lost packets: 0.24930491195551435
Ratio of corrupted packets: 0.0
Average RTT: 41830.86482864948ms
Average communication time: 52397.90473720517 ms
==================================================
```