

COMP3331 - Semester 2, Assignment 1

"Simple Transfer Protocol"

Keiran Sampson - z5168147

September 25, 2018

This protocol was designed using Python3 and is not compatible with prior versions.
The files have also been made executable, and as such can be executed by the commands:

```
python3.6 sender.py <Arguments>  
python3.6 receiver.py <Arguments>
```

or

```
./sender.py <Arguments>  
./receiver.py <Arguments>
```

Implementation and Features

I was able to successfully implement all of the STP Protocol and PLD features as outlined in the Assignment specifications. Below are overviews of the key features implemented.

sender.py

The STP Sender is divided into two main components, *listener* and *sender* threads. They work together to manage the clients network connection, file transfer, PLD and logging. See Appendix Image 1.

Listener - operated by the main thread, its primary operations include:

1. Initialising the STP Protocol:
 - a. Storing the program Arguments
 - b. Opening the file, breaking it up into chunks and queuing into MSS segments
 - c. Initialising the sending window of MWS
 - d. Connecting to the remote server using a three-segment (SYN, SYN/ACK, ACK) handshake
2. Initialising the PLD Module for use by the Sender Thread
3. Creating and starting a Sender Thread
4. Listening for incoming ACK requests:
 - a. Receive incoming segments and unpacking into a message object

5. Updating the message window:
 - a. Update log with received message stats
 - b. Update Estimated RTT and RTO
 - c. Update message window if required
 - d. If duplicate ACK:
 - i. Updating duplicate ACK count
 - ii. Carrying out Fast ReTransmission if required, and subsequently cancelling RTT calculation
 - e. Signal if file transfer completed
6. Terminating the Sender Thread
7. Closing the remote connection:
 - a. Complete a four-segment (FIN, ACK, FIN, ACK) teardown
 - b. Write final PLD Log Statistics
 - c. Program closure

Sender - operated by the spawned secondary thread, its primary operations include:

1. Calculation of Timeout Events
 - a. Logging of Timeout Events
 - b. Restarting Timeout Timer
 - c. Cancelling active RTT calculation
2. Sending new, unsent segments from the message window
3. Operating the PLD module
 - a. Calculation and execution of PLD events - ie message dropping, corruption etc
 - b. Logging of PLD events
 - c. Creating delayed message threads as required
 - d. Final transmission of segments over UDP network

receiver.py

The STP Receiver is a single threaded system. It manages the servers STP Protocol, network connections, file buffering, logging and file output. Its primary operations are:

1. Initialising the STP Protocol:
 - a. Storing the program arguments
 - b. Creating a listening UDP socket
2. Connection Management
 - a. Initialising a new client connection
 - b. Completing three-segment (SYN, SYN/ACK, ACK) handshake
 - c. Buffering out of order message segments
 - d. Managing Cumulative Acknowledgments
 - e. Managing Sequence/Ack Numbers
 - f. Sending/Receiving Messages
 - g. Completing a four-segment (FIN, ACK, FIN, ACK) teardown

3. Message Operations
 - a. Unpacking received STP segments
 - b. STP corruption checks
 - c. Packing of STP segments for sending
4. File Logging
 - a. Logging of received and sent files
 - b. Logging of STP statistics

STP HEADER

The header I designed is 88 bits in size and contains 4 fields with the variable sized payload following. The fields used are as follows:

- i. SEQ NUMBER - Increment from 0 by bytes of data (32 bits)
- ii. ACK NUMBER - Increment from 0 by bytes of data (32 bits)
- iii. FLAGS - Bitwise Flags (SYN, ACK, SYN/ACK, FIN) (8 bits)
- iv. CHECKSUM - Checksum of the STP Segment (includes payload) (16 bits)
- v. PAYLOAD - Transmitted data of variable length

See Appendix Image 2 for the Header Diagram.

Design Trade-offs and Future Improvements

Sender

The main trade off I made to the sender was to only create 2 threads. Although it does improve the sending throughput, it would be better to have implemented several producer and consumer worker threads for the sender to prevent possible bottlenecks occurring.

A listener thread would simply receive packets and add them to a worker queue. Main worker threads would then take these received messages and calculate any actions necessary (ie logging, Fast RXT etc) and update the message window. A message window thread would send any new messages in the message window, a separate timeout thread would carry out any timeout events and PLD worker threads would carry out any PLD operations before sending the messages.

Receiver

For the receiver to keep the system simple I made the significant tradeoff of not threading the system. I believe that a significant improvement to the receiver would be to implement listener, sender and worker threads (similar to the sender) to improve system throughput.

For the receiver, I have also partially implemented the ability to receive multiple files simultaneously. However because the assignment requires that we hardcode the filename to save a file under, any transmitted file is subsequently overwritten. The assignment also requires us to terminate the server upon the first files upload completing which breaks any subsequent uploads.

In my current implementation, to accommodate multiple connections, when a new connection is made I store it and any relevant information into a dictionary using the unique sender and port as the key. This allows me to accept and manage multiple connections from any sending programs at once (even if it's the same originating PC).

When a packet is received, using the sender as the key, I can retrieve the required connection information, such as its Sequence and ACK numbers, current file buffer etc and perform the operations required.

In order to be able to properly complete this system, I would need to first prevent previously uploaded files from being overwritten via a filename increment, and remove the subsequent termination of the server upon completing a file upload.

I would also change the current handshake and teardown systems. The reason for this is that they currently discard any packets that are not specifically applicable to the current stage of the handshake/teardown, reducing the systems overall efficiency should we be receiving multiple files simultaneously.

Experiments

Experiment 1 - *test0.pdf*:

Params: pDrop = 0.1, MWS = 500 bytes, MSS = 100 bytes, seed = 100, gamma = 4

To assist with understanding what has happened in the following experiments, I have added in the locations where received packets were **DROPPED**, and made the dropped sequence numbers **BOLD**.

- i. Transfer the File *test0.pdf* with pdrop = 0.1 - For result log files see Appendix Tables 1 and 2

Sequence Numbers Received by Server:

0,1,1,101,**DROP**,301,401,501,**201**,601,701,801,901,1001,1101,1201,1301,1401,1501,1601,1701,1801,1901,**DROP**,2101,2201,2301,2401,**2001**,2501,2601,**DROP,DROP**,2901,3001,**2701,2801**,3029,3030

Looking at the Sender log, we can see that 4/39 packets were dropped (approx 10% loss). However just by looking at the receiver, we can determine that this occurred where several packets arrive significantly out of sequence, ie: 2101,2201,2301,2401,2001

We would normally expect to see 2001,2101,2201,2301,2401 had no loss occurred. Of importance is that no more than 4 packets larger than the dropped packet are received before the dropped packet. This is due to the sender window containing 5 segments (including the dropped packet). Therefore it cannot slide the window up to send any new packets until the bottom (dropped) packet is acknowledged by the server.

- ii. Repeat with pdrop = 0.3 - For result log files see Appendix Tables 3 and 4

Sequence Numbers Received by Server:

0,1,**DROP**,101,201,301,1,**DROP**,501,**DROP,DROP**,801,**401,DROP**,1001,**601**,1101,**701**,1201,**901,DROP,DROP,DROP**,1601,1701,**1301**,1801,**1401,1501,DROP**,2001,2101,2201,**1901,DROP**,2401,**2301,DROP,DROP,DROP**,2801,**2501,2601,2701**,2901,3001,3029,3030

Raising pdrop to 0.3 (30%) significantly increases the number of dropped packets. Looking at the Sender log, we can determine that 24/59 packets were dropped. Approx 41%. The dramatic increase in the number of dropped packets has significantly slowed down the file upload process, now taking 21 seconds to complete.

Note however that the receiver only observed 13 dropped packets. The reason for this is that the receiver does not know when several sequential packets are dropped by the sender in a row (with no subsequent packets being successfully sent). This results in the sender experiencing a time-out. When a timeout is followed by another dropped packet, it results in even further time delays.

Experiment 2 - test1.pdf:

Params: pdrop = 0.5, MWS = 500 bytes, MSS = 50 bytes, seed = 300, pdelay = 0.2, MaxDelay = 1000
*TimeoutInterval = EstimatedRTT + gamma * DevRTT*

During this experiment we ran the same file transfer 3 times with increasing gamma values: 2, 4 and 6.

We observed that increasing from 2 to 4/6 significantly affected the TimeoutInterval calculations causing the program to wait increasingly longer for any timeouts and subsequent retransmissions caused by dropped/delayed packets. This resulted in the observed file transmission time increases whenever the gamma value was increased. However, the benefit of this increased timeout value was a reduction in the total number of segment retransmissions. This is because delayed segments now had time to be received by the server and subsequently ACK'd before the server timed out.

The reason that gamma 6 does not see any real change in the number of total transmitted segments, but is still significantly slower than gamma 4 is that delayed packets are already being ACK'd before any timer timeouts in both cases. However, dropped packets are now significantly slowing the file transfer as the protocol is waiting significantly longer to timeout and retransmit than it did for gamma 4.

- i. gamma = 2 - For result log files see Appendix Tables 6 and 7
- ii. gamma = 4 - For result log files see Appendix Tables 8 and 9
- iii. gamma = 6 - For result log files see Appendix Tables 10 and 11

Note that only the start and end of the log files are included due to their significant size.

Experiment 3 - test2.pdf:

Params: MWS = 500, MSS = 50, gamma = 4, pDrop = 0.1, pDuplicate = 0.1, pCorrupt = 0.1, pOrder = 0.1, maxOrder = 4, pDelay = 0, maxDelay = 0, and seed = 300

The transfer of this file took 58 minutes 29 seconds.

To determine the worst cause of the file transfer delay I looked at each event as a percentage of the total segments transmitted via the PLD. See Appendix Table 12 for these results.

From this we can observe that there were more dropped packets than any other event. When a packet is dropped or corrupted, the best scenario is that there is a fast retransmission, worst scenario is we have to wait for a timeout. This causes significant delays, and if the segment dropped was the smallest segment in the transmission window, stops the file upload until it is successfully received by the receiver.

Duplicate packets, are unhelpful as they are wasted bandwidth, but are not stopping the file upload from progressing. Similarly out of order packets cause fast retransmissions and subsequently duplicate packets, but because we are using cumulative acknowledgements, they also don't prevent the file upload from progressing.

Appendix

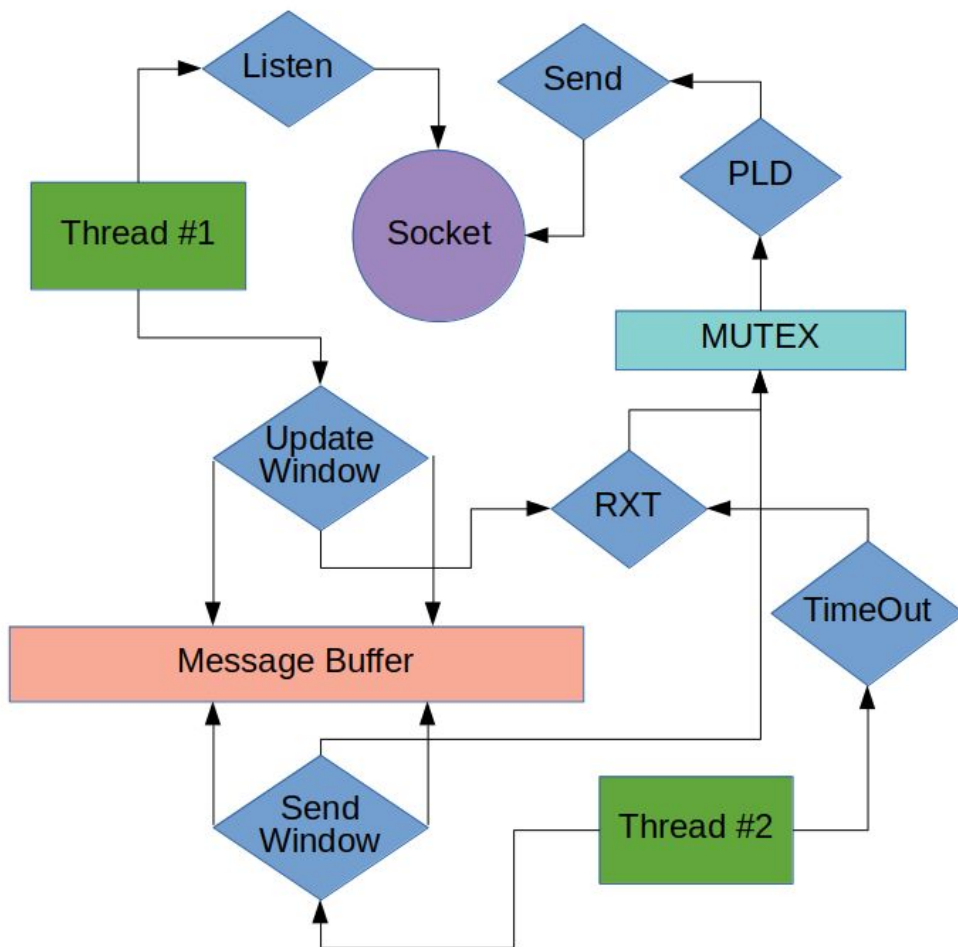


Image 1: Sender System Overview

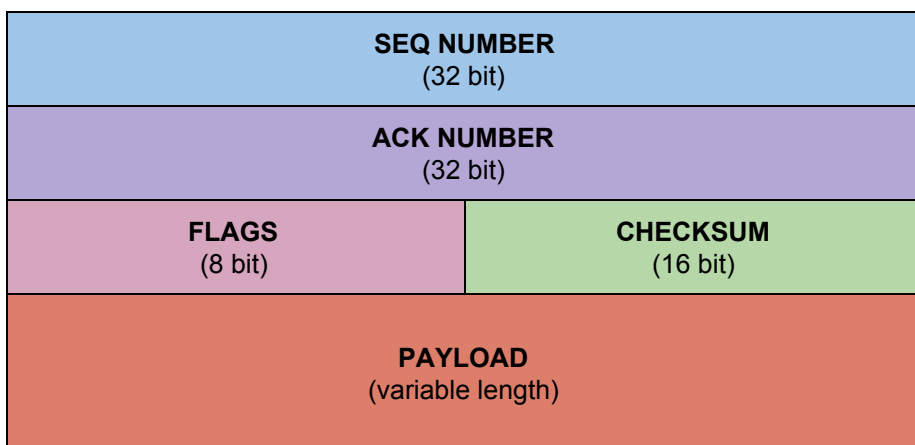


Image 2: STP Message Header

rcv 0.76 S 0 0 0 snd 0.76 SA 0 0 1 rcv 0.76 A 1 0 1 rcv 0.76 D 1 100 1 snd 0.76 A 1 0 101 rcv 0.76 D 101 100 1 snd 0.76 A 1 0 201 rcv 0.78 D 301 100 1 snd/DA 0.78 A 1 0 201 rcv 0.78 D 401 100 1 snd/DA 0.78 A 1 0 201 rcv 0.78 D 501 100 1 snd/DA 0.78 A 1 0 201 rcv 0.78 D 201 100 1 snd 0.78 A 1 0 601 rcv 0.78 D 601 100 1 snd 0.78 A 1 0 701 rcv 0.87 D 701 100 1 snd 0.87 A 1 0 801 rcv 0.87 D 801 100 1 snd 0.87 A 1 0 901 rcv 0.87 D 901 100 1 snd 0.87 A 1 0 1001 rcv 0.87 D 1001 100 1 snd 0.87 A 1 0 1101 rcv 0.87 D 1101 100 1 snd 0.87 A 1 0 1201 rcv 0.87 D 1201 100 1 snd 0.87 A 1 0 1301 rcv 0.87 D 1301 100 1 snd 0.87 A 1 0 1401 rcv 0.87 D 1401 100 1 snd 0.87 A 1 0 1501 rcv 0.87 D 1501 100 1 snd 0.87 A 1 0 1601 rcv 0.87 D 1601 100 1 snd 0.87 A 1 0 1701 rcv 0.87 D 1701 100 1 snd 0.87 A 1 0 1801 rcv 0.87 D 1801 100 1	snd 0.87 A 1 0 1901 rcv 0.87 D 1901 100 1 snd 0.87 A 1 0 2001 rcv 0.87 D 2101 100 1 snd/DA 0.87 A 1 0 2001 rcv 0.87 D 2201 100 1 snd/DA 0.87 A 1 0 2001 rcv 0.87 D 2301 100 1 snd/DA 0.87 A 1 0 2001 rcv 0.87 D 2401 100 1 snd/DA 0.87 A 1 0 2001 rcv 0.87 D 2001 100 1 snd 0.87 A 1 0 2501 rcv 1.02 D 2501 100 1 snd 1.02 A 1 0 2601 rcv 1.02 D 2601 100 1 snd 1.02 A 1 0 2701 rcv 1.02 D 2901 100 1 snd/DA 1.02 A 1 0 2701 rcv 1.02 D 3001 28 1 snd/DA 1.02 A 1 0 2701 rcv 1.06 D 2701 100 1 snd 1.06 A 1 0 2801 rcv 2.82 D 2801 100 1 snd 2.82 A 1 0 3029 rcv 2.87 F 3029 0 1 snd 2.87 A 1 0 3030 snd 2.87 F 1 0 3030 rcv 2.87 A 3030 0 2 =====	Amount of data received (bytes) 3028 Total Segments Received 35 Data segments received 31 Data segments with Bit Errors 0 Duplicate data segments received 0 Duplicate ACKs sent 9 =====	snd 0.0 S 0 0 0 rcv 0.0 SA 0 0 1 snd 0.0 A 1 0 1 snd 0.0 D 1 100 1 rcv 0.0 A 1 0 101 snd 0.0 D 101 100 1 rcv 0.0 A 1 0 201 snd/drop 0.0 D 201 100 1 snd 0.02 D 301 100 1 rcv/DA 0.02 A 1 0 201 snd 0.02 D 401 100 1 rcv/DA 0.02 A 1 0 201 snd 0.02 D 501 100 1 snd/RXT 0.02 D 201 100 1 snd 0.02 D 601 100 1 rcv/DA 0.03 A 1 0 201 rcv 0.08 A 1 0 601 rcv 0.11 A 1 0 701 snd 0.11 D 701 100 1 rcv 0.11 A 1 0 801 snd 0.11 D 801 100 1 rcv 0.11 A 1 0 901 snd 0.11 D 901 100 1 rcv 0.11 A 1 0 1001 snd 0.11 D 1001 100 1 rcv 0.11 A 1 0 1101 snd 0.11 D 1101 100 1 rcv 0.11 A 1 0 1201 snd 0.11 D 1201 100 1 snd 0.11 D 1301 100 1 rcv 0.11 A 1 0 1301 snd 0.11 D 1401 100 1 rcv 0.11 A 1 0 1401 snd 0.11 D 1501 100 1 rcv 0.11 A 1 0 1501 snd 0.11 D 1601 100 1 rcv 0.11 A 1 0 1601 snd 0.11 D 1701 100 1 rcv 0.11 A 1 0 1701 snd 0.11 D 1801 100 1 rcv 0.11 A 1 0 1801 snd 0.11 D 1901 100 1 rcv 0.11 A 1 0 1901 snd/drop 0.11 D 2001 100 1 rcv 0.11 A 1 0 2001 snd 0.11 D 2101 100 1 snd 0.11 D 2201 100 1	rcv/DA 0.11 A 1 0 2001 snd 0.12 D 2301 100 1 rcv/DA 0.12 A 1 0 2001 snd 0.12 D 2401 100 1 snd/RXT 0.12 D 2001 100 1 rcv/DA 0.15 A 1 0 2001 rcv/DA 0.19 A 1 0 2001 rcv 0.23 A 1 0 2501 snd 0.26 D 2501 100 1 snd 0.26 D 2601 100 1 snd/drop 0.26 D 2701 100 1 rcv 0.26 A 1 0 2601 snd/drop 0.26 D 2801 100 1 rcv 0.26 A 1 0 2701 snd 0.26 D 2901 100 1 rcv/DA 0.26 A 1 0 2701 snd 0.26 D 3001 28 1 rcv/DA 0.26 A 1 0 2701 snd/RXT 0.3 D 2701 100 1 rcv 0.34 A 1 0 2801 snd/RXT 2.06 D 2801 100 1 rcv 2.07 A 1 0 3029 snd 2.11 F 3029 0 1 rcv 2.11 A 1 0 3030 rcv 2.11 F 1 0 3030 snd 2.11 A 3030 0 2 =====	Size of the file (in Bytes) 3028 Segments transmitted (including drop & RXT) 39 Number of Segments handled by PLD 35 Number of Segments dropped 4 Number of Segments Corrupted 0 Number of Segments Re-ordered 0 Number of Segments Duplicated 0 Number of Segments Delayed 0 Number of Retransmissions due to TIMEOUT 1 Number of FAST RETRANSMISSION 3 Number of DUP ACKS received 9 =====
--	---	--	--	---	---

Table 1: Experiment 1 (i) Receiver Log File

Table 2: Experiment 1 (i) Sender Log File

rcv 0.48 S 0 0 0 snd 0.48 SA 0 0 1 rcv 0.48 A 1 0 1 rcv 0.48 D 101 100 1 snd/DA 0.48 A 1 0 1 rcv 0.48 D 201 100 1 snd/DA 0.48 A 1 0 1 rcv 0.48 D 301 100 1 snd/DA 0.48 A 1 0 1 rcv 3.49 D 1 100 1 snd 3.49 A 1 0 401 rcv 3.52 D 501 100 1 snd/DA 3.52 A 1 0 401 rcv 3.52 D 801 100 1 snd/DA 3.52 A 1 0 401 rcv 3.55 D 401 100 1 snd 3.55 A 1 0 601 rcv 3.63 D 1001 100 1 snd/DA 3.63 A 1 0 601 rcv 4.62 D 601 100 1 snd 4.62 A 1 0 701 rcv 4.65 D 1101 100 1 snd/DA 4.65 A 1 0 701 rcv 5.65 D 701 100 1 snd 5.65 A 1 0 901 rcv 5.68 D 1201 100 1 snd/DA 5.68 A 1 0 901 rcv 6.68 D 901 100 1 snd 6.68 A 1 0 1301 rcv 6.78 D 1601 100 1 snd/DA 6.78 A 1 0 1301 rcv 6.78 D 1701 100 1 snd/DA 6.78 A 1 0 1301 rcv 7.86 D 1301 100 1 snd 7.86 A 1 0 1401 rcv 7.88 D 1801 100 1 snd/DA 7.88 A 1 0 1401 rcv 8.88 D 1401 100 1 snd 8.88 A 1 0 1501 rcv 13.92 D 1501 100 1 snd 13.92 A 1 0 1901 rcv 13.93 D 2001 100 1 snd/DA 13.93 A 1 0 1901 rcv 13.93 D 2101 100 1 snd/DA 13.93 A 1 0 1901 rcv 13.93 D 2201 100 1	snd/DA 13.93 A 1 0 1901 rcv 13.93 D 1901 100 1 snd 13.93 A 1 0 2301 rcv 13.96 D 2401 100 1 snd/DA 13.96 A 1 0 2301 rcv 14.93 D 2301 100 1 snd 14.93 A 1 0 2501 rcv 14.97 D 2801 100 1 snd/DA 14.97 A 1 0 2501 rcv 15.97 D 2501 100 1 snd 15.97 A 1 0 2601 rcv 16.98 D 2601 100 1 snd 16.98 A 1 0 2701 rcv 18.03 D 2701 100 1 snd 18.03 A 1 0 2901 rcv 20.03 D 2901 100 1 snd 20.03 A 1 0 3001 rcv 22.04 D 3001 28 1 snd 22.04 A 1 0 3029 rcv 22.06 F 3029 0 1 snd 22.06 A 1 0 3030 snd 22.06 F 1 0 3030 rcv 22.06 A 3030 0 2 ===== Amount of data received (bytes) 3028 Total Segments Received 35 Data segments received 31 Data segments with Bit Errors 0 Duplicate data segments received 0 Duplicate ACKs sent 16 =====	snd 0.0 S 0 0 0 rcv 0.0 SA 0 0 1 snd 0.0 A 1 0 1 snd/drop 0.0 D 1 100 1 snd 0.0 D 101 100 1 rcv 0.0 A 1 0 1 snd 0.0 D 201 100 1 snd 0.0 D 301 100 1 snd/drop 0.0 D 401 100 1 rcv 0.0 A 1 0 1 rcv 0.03 A 1 0 1 snd/RXT/drop 1.0 D 1 100 1 snd/RXT/drop 2.01 D 1 100 1 snd/RXT 3.01 D 1 100 1 rcv 3.01 A 1 0 401 snd 3.04 D 501 100 1 snd/drop 3.04 D 601 100 1 snd/drop 3.04 D 701 100 1 rcv/DA 3.04 A 1 0 401 snd 3.04 D 801 100 1 rcv/DA 3.04 A 1 0 401 snd/RXT 3.07 D 401 100 1 rcv 3.11 A 1 0 601 snd/drop 3.14 D 901 100 1 snd 3.15 D 1001 100 1 rcv/DA 3.16 A 1 0 601 snd/RXT 4.14 D 601 100 1 rcv 4.14 A 1 0 701 snd 4.17 D 1101 100 1 rcv/DA 4.17 A 1 0 701 snd/RXT 5.17 D 701 100 1 rcv 5.18 A 1 0 901 snd 5.2 D 1201 100 1 snd/drop 5.2 D 1301 100 1 rcv/DA 5.2 A 1 0 901 snd/RXT 6.2 D 901 100 1 rcv 6.2 A 1 0 1301 snd/drop 6.23 D 1401 100 1 snd/drop 6.27 D 1501 100 1 snd 6.3 D 1601 100 1 rcv/DA 6.3 A 1 0 1301 snd 6.3 D 1701 100 1 rcv/DA 6.31 A 1 0 1301 snd/RXT/drop 6.35 D 1301 100 1 snd/RXT 7.38 D 1301 100 1 rcv 7.39 A 1 0 1401 snd 7.41 D 1801 100 1 rcv/DA 7.41 A 1 0 1401 snd/RXT 8.41 D 1401 100 1 rcv 8.42 A 1 0 1501 snd/drop 8.44 D 1901 100 1 snd/RXT/drop 9.44 D 1501 100 1 snd/RXT/drop 10.44 D 1501 100 1 snd/RXT/drop 11.44 D 1501 100 1 snd/RXT/drop 12.44 D 1501 100 1 snd/RXT 13.44 D 1501 100 1	rcv 13.44 A 1 0 1901 snd 13.45 D 2001 100 1 rcv/DA 13.45 A 1 0 1901 snd 13.45 D 2101 100 1 rcv/DA 13.45 A 1 0 1901 snd 13.45 D 2201 100 1 snd/RXT 13.45 D 1901 100 1 rcv/DA 13.46 A 1 0 1901 rcv 13.46 A 1 0 2301 snd/drop 13.45 D 2301 100 1 rcv/DA 13.48 A 1 0 2301 snd 13.47 D 2401 100 1 snd/drop 13.48 D 2501 100 1 snd/drop 13.48 D 2601 100 1 snd/drop 13.48 D 2701 100 1 snd/RXT 14.46 D 2301 100 1 rcv 14.46 A 1 0 2501 snd 14.49 D 2801 100 1 snd/drop 14.49 D 2901 100 1 rcv/DA 14.49 A 1 0 2501 snd/RXT 15.49 D 2501 100 1 rcv 15.49 A 1 0 2601 snd/drop 15.5 D 3001 28 1 snd/RXT 16.5 D 2601 100 1 rcv 16.51 A 1 0 2701 snd/RXT 17.55 D 2701 100 1 rcv 17.55 A 1 0 2901 snd/RXT/drop 18.55 D 2901 100 1 rcv 19.56 A 1 0 3001 snd/RXT 19.56 D 2901 100 1 snd/RXT/drop 20.56 D 3001 28 1 snd/RXT 21.56 D 3001 28 1 rcv 21.56 A 1 0 3029 snd 21.58 F 3029 0 1 rcv 21.58 A 1 0 3030 rcv 21.58 F 1 0 3030 snd 21.58 A 3030 0 2 ===== Size of the file (in Bytes) 3028 Segments transmitted (including drop & RXT) 59 Number of Segments handled by PLD 55 Number of Segments dropped 24 Number of Segments Corrupted 0 Number of Segments Re-ordered 0 Number of Segments Duplicated 0 Number of Segments Delayed 0 Number of Retransmissions due to TIMEOUT 21 Number of FAST RETRANSMISSION 3 Number of DUP ACKS received 13 =====
--	---	--	--

Table 3: Experiment 1 (ii) Receiver Log File

Table 4: Experiment 1 (ii) Sender Log File

Gamma	Execution Time	Segments Transmitted
2	2686.12	13155
4	10061.16	12450
6	13553.95	12459

Table 5: Experiment 2 Gamma Results

Start Receiver Log	End Receiver Log	Start Sender Log	End Sender Log
rcv 1.45 S 0 0 0 snd 1.45 SA 0 0 1 rcv 1.45 A 1 0 1 rcv 2.0 D 1 50 1 snd 2.0 A 1 0 51 rcv 2.01 D 201 50 1 snd/DA 2.01 A 1 0 51 rcv 2.01 D 301 50 1 snd/DA 2.01 A 1 0 51 rcv 2.01 D 51 50 1 snd 2.01 A 1 0 101 rcv 2.01 D 401 50 1 snd/DA 2.01 A 1 0 101 rcv 2.01 D 501 50 1 snd/DA 2.01 A 1 0 101 rcv 2.02 D 101 50 1 snd 2.02 A 1 0 151 rcv 2.31 D 101 50 1 snd/DA 2.31 A 1 0 151 rcv 2.93 D 451 50 1 snd/DA 2.93 A 1 0 151 rcv 2.97 D 151 50 1 snd 2.98 A 1 0 251 rcv 5.17 D 251 50 1 snd 5.17 A 1 0 351 rcv 6.28 D 351 50 1 snd 6.28 A 1 0 551 rcv 6.33 D 1001 50 1 snd/DA 6.33 A 1 0 551 rcv 11.66 D 551 50 1 snd 11.66 A 1 0 601 rcv 12.99 D 601 50 1 snd 12.99 A 1 0 651 rcv 16.47 D 651 50 1 snd 16.47 A 1 0 701 rcv 17.58 D 701 50 1 snd 17.58 A 1 0 751 rcv 17.62 D 1201 50 1 snd/DA 17.62 A 1 0 751 (SKIP LOG UNTIL END)	snd 2681.24 A 1 0 307501 rcv 2681.77 D 307501 50 1 snd 2681.77 A 1 0 307601 rcv 2682.79 D 307601 50 1 snd 2682.79 A 1 0 307651 rcv 2683.14 D 307651 50 1 snd 2683.14 A 1 0 307701 rcv 2683.24 D 308151 50 1 snd/DA 2683.24 A 1 0 307701 rcv 2683.34 D 307701 50 1 snd 2683.34 A 1 0 307801 rcv 2683.38 D 308201 3 1 snd/DA 2683.38 A 1 0 307801 rcv 2683.41 D 308101 50 1 snd/DA 2683.41 A 1 0 307801 rcv 2684.63 D 307801 50 1 snd 2684.63 A 1 0 307851 rcv 2684.83 D 307851 50 1 snd 2684.84 A 1 0 307901 rcv 2685.04 D 307901 50 1 snd 2685.05 A 1 0 307951 rcv 2685.25 D 307951 50 1 snd 2685.25 A 1 0 308001 rcv 2686.44 D 308001 50 1 snd 2686.44 A 1 0 308051 rcv 2687.52 D 308051 50 1 snd 2687.52 A 1 0 308204 rcv 2687.57 F 308204 0 1 snd 2687.57 A 1 0 308205 rcv 2687.57 F 1 0 308205 rcv 2687.57 A 308205 0 2 ===== Amount of data received (bytes) 319581 Total Segments Received 6526 Data segments received 6522 Data segments with Bit Errors 0 Duplicate data segments received 357 Duplicate ACKs sent 3425 =====	snd 0.0 S 0 0 0 rcv 0.0 SA 0 0 1 snd 0.0 A 1 0 1 snd 0.56 D 1 50 1 snd/drop 0.56 D 51 50 1 rcv 0.56 A 1 0 51 snd/drop 0.56 D 151 50 1 snd 0.56 D 201 50 1 snd/drop 0.56 D 251 50 1 rcv/DA 0.56 A 1 0 51 snd 0.56 D 301 50 1 rcv/DA 0.56 A 1 0 51 snd/drop 0.56 D 351 50 1 snd/RXT 0.56 D 51 50 1 snd 0.56 D 401 50 1 rcv 0.56 A 1 0 101 snd 0.56 D 501 50 1 rcv/DA 0.56 A 1 0 101 snd/drop 0.56 D 551 50 1 rcv/DA 0.56 A 1 0 101 snd/RXT 0.57 D 101 50 1 rcv 0.6 A 1 0 151 snd/drop 0.63 D 601 50 1 snd/dely 0.87 D 101 50 1 rcv/DA 0.89 A 1 0 151 snd/dely 1.49 D 451 50 1 rcv/DA 1.51 A 1 0 151 snd/RXT 1.53 D 151 50 1 rcv 1.56 A 1 0 251 snd/drop 1.58 D 651 50 1 snd/drop 1.58 D 701 50 1 snd/RXT/drop 2.65 D 251 50 1 snd/RXT 3.72 D 251 50 1 rcv 3.73 A 1 0 351 snd/drop 3.76 D 751 50 1 snd/drop 3.76 D 801 50 1 snd/RXT 4.83 D 351 50 1 rcv 4.83 A 1 0 551 (SKIP LOG UNTIL END)	rcv 2683.19 A 1 0 307851 snd/RXT 2683.38 D 307851 50 1 rcv 2683.4 A 1 0 307901 snd/RXT 2683.59 D 307901 50 1 rcv 2683.61 A 1 0 307951 snd/RXT 2683.8 D 307951 50 1 rcv 2683.81 A 1 0 308001 snd/RXT/drop 2684.0 D 308001 50 1 snd/RXT/drop 2684.17 D 308001 50 1 snd/RXT/drop 2684.33 D 308001 50 1 snd/RXT/drop 2684.49 D 308001 50 1 snd/RXT/drop 2684.66 D 308001 50 1 snd/RXT/drop 2684.82 D 308001 50 1 snd/RXT 2684.99 D 308001 50 1 rcv 2684.99 A 1 0 308051 snd/RXT/drop 2685.18 D 308051 50 1 snd/RXT/drop 2685.35 D 308051 50 1 snd/RXT/drop 2685.51 D 308051 50 1 snd/RXT/drop 2685.84 D 308051 50 1 snd/RXT/drop 2686.01 D 308051 50 1 snd/dely 2686.07 D 308051 50 1 rcv 2686.09 A 1 0 308204 snd 2686.12 F 308204 0 1 rcv 2686.12 A 1 0 308205 rcv 2686.12 F 1 0 308205 snd 2686.12 A 308205 0 2 ===== Size of the file (in Bytes) 308203 Segments transmitted (including drop & RXT) 13155 Number of Segments handled by PLD 13151 Number of Segments dropped 6629 Number of Segments Corrupted 0 Number of Segments Re-ordered 0 Number of Segments Duplicated 0 Number of Segments Delayed 1296 Number of Retransmissions due to TIMEOUT 6063 Number of FAST RETRANSMISSION 923 Number of DUP ACKS received 3425 =====

Table 6: Experiment 2 (i) Receiver Log File

Table 7: Experiment 2 (i) Sender Log File

Start Receiver Log	End Receiver Log	Start Sender Log	End Sender Log
rcv 1.58 S 0 0 0 snd 1.58 SA 0 0 1 rcv 1.58 A 1 0 1 rcv 2.13 D 1 50 1 snd 2.13 A 1 0 51 rcv 2.18 D 201 50 1 snd/DA 2.18 A 1 0 51 rcv 2.21 D 301 50 1 snd/DA 2.22 A 1 0 51 rcv 2.22 D 401 50 1 snd/DA 2.22 A 1 0 51 rcv 2.22 D 51 50 1 snd 2.22 A 1 0 101 rcv 2.22 D 501 50 1 snd/DA 2.22 A 1 0 101 rcv 2.44 D 101 50 1 snd 2.44 A 1 0 151 rcv 2.48 D 601 50 1 snd/DA 2.48 A 1 0 151 rcv 3.16 D 451 50 1 snd/DA 3.17 A 1 0 151 rcv 4.92 D 151 50 1 snd 4.92 A 1 0 251 rcv 8.34 D 251 50 1 snd 8.34 A 1 0 351 rcv 10.05 D 351 50 1 snd 10.05 A 1 0 551 rcv 10.19 D 1001 50 1 snd/DA 10.19 A 1 0 551 (SKIP LOG UNTIL END)	rcv 10050.53 D 307301 50 1 snd 10050.53 A 1 0 307351 rcv 10050.58 D 307801 50 1 snd/DA 10050.58 A 1 0 307351 rcv 10053.95 D 307351 50 1 snd 10053.96 A 1 0 307451 rcv 10054.0 D 307851 50 1 snd/DA 10054.0 A 1 0 307451 rcv 10054.01 D 307901 50 1 snd/DA 10054.01 A 1 0 307451 rcv 10054.05 D 307451 50 1 snd 10054.05 A 1 0 307701 rcv 10054.12 D 308051 50 1 snd/DA 10054.12 A 1 0 307701 rcv 10054.13 D 308101 50 1 snd/DA 10054.13 A 1 0 307701 rcv 10054.13 D 308151 50 1 snd/DA 10054.13 A 1 0 307701 rcv 10054.2 D 307951 50 1 snd/DA 10054.21 A 1 0 307701 rcv 10057.53 D 307701 50 1 snd 10057.53 A 1 0 308001 rcv 10060.96 D 308001 50 1 snd 10060.96 A 1 0 308201 rcv 10062.71 D 308201 3 1 snd 10062.71 A 1 0 308204 rcv 10062.73 F 308204 0 1 snd 10062.73 A 1 0 308205 snd 10062.74 F 1 0 308205 rcv 10062.74 A 308205 0 2 ===== Amount of data received (bytes) 308303 Total Segments Received 6171 Data segments received 6167 Data segments with Bit Errors 0 Duplicate data segments received 2 Duplicate ACKs sent 3042 =====	snd 0.0 S 0 0 0 rcv 0.0 SA 0 0 1 snd 0.0 A 1 0 1 snd 0.56 D 1 50 1 rcv 0.56 A 1 0 51 snd/drop 0.56 D 51 50 1 snd/drop 0.59 D 151 50 1 snd 0.6 D 201 50 1 rcv/DA 0.6 A 1 0 51 snd/drop 0.6 D 251 50 1 snd 0.64 D 301 50 1 snd/drop 0.64 D 351 50 1 rcv/DA 0.64 A 1 0 51 snd 0.64 D 401 50 1 snd/RXT 0.64 D 51 50 1 snd 0.64 D 501 50 1 rcv/DA 0.64 A 1 0 51 rcv 0.66 A 1 0 101 snd/drop 0.69 D 551 50 1 rcv/DA 0.69 A 1 0 101 snd/dely 0.87 D 101 50 1 rcv 0.87 A 1 0 151 snd 0.9 D 601 50 1 rcv/DA 0.9 A 1 0 151 snd/dely 1.59 D 451 50 1 rcv/DA 1.6 A 1 0 151 snd/RXT/drop 1.63 D 151 50 1 snd/RXT 3.34 D 151 50 1 rcv 3.35 A 1 0 251 snd/drop 3.38 D 651 50 1 snd/drop 3.4 D 701 50 1 snd/RXT/drop 5.05 D 251 50 1 snd/RXT 6.76 D 251 50 1 rcv 6.76 A 1 0 351 (SKIP LOG UNTIL END)	snd/RXT 10052.38 D 307351 50 1 rcv 10052.39 A 1 0 307451 snd 10052.42 D 307851 50 1 rcv/DA 10052.43 A 1 0 307451 snd 10052.43 D 307901 50 1 rcv/DA 10052.44 A 1 0 307451 snd/RXT 10052.47 D 307451 50 1 rcv 10052.5 A 1 0 307701 snd/drop 10052.54 D 308001 50 1 snd 10052.54 D 308051 50 1 rcv/DA 10052.55 A 1 0 307701 snd 10052.55 D 308101 50 1 rcv/DA 10052.55 A 1 0 307701 snd 10052.55 D 308151 50 1 snd/RXT/drop 10052.56 D 307701 50 1 rcv/DA 10052.57 A 1 0 307701 snd/dely 10052.63 D 307951 50 1 rcv/DA 10052.64 A 1 0 307701 snd/RXT/drop 10054.26 D 307701 50 1 snd/RXT 10055.95 D 307701 50 1 rcv 10055.96 A 1 0 308001 snd/drop 10056.0 D 308201 3 1 snd/RXT/drop 10057.69 D 308001 50 1 snd/RXT 10059.38 D 308001 50 1 rcv 10059.39 A 1 0 308201 snd/RXT 10061.13 D 308201 3 1 rcv 10061.14 A 1 0 308204 snd 10061.16 F 308204 0 1 rcv 10061.16 A 1 0 308205 rcv 10061.16 F 1 0 308205 snd 10061.16 A 308205 0 2 ===== Size of the file (in Bytes) 308203 Segments transmitted (including drop & RXT) 12450 Number of Segments handled by PLD 12446 Number of Segments dropped 6279 Number of Segments Corrupted 0 Number of Segments Re-ordered 0 Number of Segments Duplicated 0 Number of Segments Delayed 1234 Number of Retransmissions due to TIMEOUT 5572 Number of FAST RETRANSMISSION 709 Number of DUP ACKS received 3042 =====

Table 8: Experiment 2 (ii) Receiver Log File

Table 9: Experiment 2 (ii) Sender Log File

Start Receiver Log	End Receiver Log	Start Sender Log	End Sender Log
rcv 2.99 S 0 0 0 snd 3.0 SA 0 0 1 rcv 3.0 A 1 0 1 rcv 3.55 D 1 50 1 snd 3.55 A 1 0 51 rcv 3.6 D 201 50 1 snd/DA 3.6 A 1 0 51 rcv 3.63 D 301 50 1 snd/DA 3.63 A 1 0 51 rcv 3.63 D 51 50 1 snd 3.63 A 1 0 101 rcv 3.63 D 401 50 1 snd/DA 3.63 A 1 0 101 rcv 3.63 D 501 50 1 snd/DA 3.63 A 1 0 101 rcv 3.65 D 101 50 1 snd 3.65 A 1 0 151 rcv 3.86 D 101 50 1 snd/DA 3.86 A 1 0 151 rcv 4.55 D 451 50 1 snd/DA 4.55 A 1 0 151 rcv 4.59 D 151 50 1 snd 4.59 A 1 0 251 rcv 9.34 D 251 50 1 snd 9.34 A 1 0 351 rcv 11.67 D 351 50 1 snd 11.68 A 1 0 551 rcv 11.82 D 1001 50 1 snd/DA 11.82 A 1 0 551 rcv 23.38 D 551 50 1 snd 23.38 A 1 0 601 rcv 25.97 D 601 50 1 snd 25.98 A 1 0 651 (SKIP LOG UNTIL END)	rcv 13534.63 D 307701 50 1 snd/DA 13534.63 A 1 0 307251 rcv 13534.69 D 307251 50 1 snd 13534.69 A 1 0 307501 rcv 13534.84 D 307851 50 1 snd/DA 13534.84 A 1 0 307501 rcv 13534.84 D 307751 50 1 snd/DA 13534.84 A 1 0 307501 rcv 13534.85 D 307901 50 1 snd/DA 13534.85 A 1 0 307501 rcv 13534.85 D 307951 50 1 snd/DA 13534.85 A 1 0 307501 rcv 13539.53 D 307501 50 1 snd 13539.55 A 1 0 307801 rcv 13539.64 D 308101 50 1 snd/DA 13539.65 A 1 0 307801 rcv 13539.66 D 308151 50 1 snd/DA 13539.67 A 1 0 307801 rcv 13539.68 D 308201 3 1 snd/DA 13539.69 A 1 0 307801 rcv 13542.13 D 307801 50 1 snd 13542.14 A 1 0 308001 rcv 13551.87 D 308001 50 1 snd 13551.88 A 1 0 308051 rcv 13556.87 D 308051 50 1 snd 13556.87 A 1 0 308204 rcv 13556.94 F 308204 0 1 snd 13556.94 A 1 0 308205 snd 13556.94 F 1 0 308205 rcv 13556.94 A 308205 0 2 ===== Amount of data received (bytes) 308503 Total Segments Received 6175 Data segments received 6171 Data segments with Bit Errors 0 Duplicate data segments received 6 Duplicate ACKs sent 3040 =====	snd 0.0 S 0 0 0 rcv 0.0 SA 0 0 1 snd 0.0 A 1 0 1 snd 0.56 D 1 50 1 rcv 0.56 A 1 0 51 snd/drop 0.56 D 51 50 1 snd/drop 0.59 D 151 50 1 snd 0.6 D 201 50 1 rcv/DA 0.61 A 1 0 51 snd/drop 0.61 D 251 50 1 snd 0.63 D 301 50 1 rcv/DA 0.63 A 1 0 51 snd/drop 0.63 D 351 50 1 snd/RXT 0.63 D 51 50 1 snd 0.63 D 401 50 1 rcv 0.63 A 1 0 101 snd 0.64 D 501 50 1 rcv/DA 0.64 A 1 0 101 snd/drop 0.64 D 551 50 1 rcv/DA 0.64 A 1 0 101 snd/RXT 0.65 D 101 50 1 rcv 0.69 A 1 0 151 snd/drop 0.73 D 601 50 1 snd/dely 0.87 D 101 50 1 rcv/DA 0.89 A 1 0 151 snd/dely 1.56 D 451 50 1 rcv/DA 1.57 A 1 0 151 snd/RXT 1.6 D 151 50 1 rcv 1.63 A 1 0 251 snd/drop 1.67 D 651 50 1 snd/drop 1.69 D 701 50 1 snd/RXT/drop 3.98 D 251 50 1 snd/RXT 6.33 D 251 50 1 (SKIP LOG UNTIL END)	snd/RXT 13536.54 D 307501 50 1 rcv 13536.56 A 1 0 307801 snd/drop 13536.6 D 308001 50 1 snd/drop 13536.62 D 308051 50 1 snd 13536.64 D 308101 50 1 rcv/DA 13536.66 A 1 0 307801 snd 13536.66 D 308151 50 1 rcv/DA 13536.67 A 1 0 307801 snd 13536.68 D 308201 3 1 snd/RXT/drop 13536.69 D 307801 50 1 rcv/DA 13536.72 A 1 0 307801 snd/dely 13539.14 D 307801 50 1 rcv 13539.16 A 1 0 308001 snd/RXT/drop 13541.5 D 308001 50 1 snd/RXT/drop 13543.84 D 308001 50 1 snd/RXT/drop 13546.16 D 308001 50 1 snd/dely 13548.88 D 308001 50 1 rcv 13548.91 A 1 0 308051 snd/RXT/drop 13551.26 D 308051 50 1 snd/dely 13553.88 D 308051 50 1 rcv 13553.9 A 1 0 308204 snd 13553.95 F 308204 0 1 rcv 13553.95 A 1 0 308205 rcv 13553.95 F 1 0 308205 snd 13553.95 A 308205 0 2 ===== Size of the file (in Bytes) 308203 Segments transmitted (including drop & RXT) 12459 Number of Segments handled by PLD 12455 Number of Segments dropped 6284 Number of Segments Corrupted 0 Number of Segments Re-ordered 0 Number of Segments Duplicated 0 Number of Segments Delayed 1237 Number of Retransmissions due to TIMEOUT 5575 Number of FAST RETRANSMISSION 715 Number of DUP ACKS received 3040 =====

Table 10: Experiment 2 (iii) Receiver Log File

Table 11: Experiment 2 (iii) Sender Log File

Start Receiver Log	End Receiver Log	Start Sender Log	End Sender Log
rcv 7.03 S 0 0 0 snd 7.03 SA 0 0 1 rcv 7.03 A 1 0 1 rcv 21.75 D 51 50 1 snd/DA 21.75 A 1 0 1 rcv 21.77 D 101 50 1 snd/DA 21.77 A 1 0 1 rcv 21.78 D 151 50 1 snd/DA 21.78 A 1 0 1 rcv 21.79 D 201 50 1 snd/DA 21.79 A 1 0 1 rcv 21.8 D 251 50 1 snd/DA 21.8 A 1 0 1 rcv 21.8 D 251 50 1 snd/DA 21.8 A 1 0 1 rcv 21.81 D 301 50 1 snd/DA 21.81 A 1 0 1 rcv 21.84 D 401 50 1 snd/DA 21.84 A 1 0 1 rcv 21.85 D 451 50 1 snd/DA 21.85 A 1 0 1 rcv 22.75 D 1 50 1 snd 22.75 A 1 0 351 rcv 22.76 D 501 50 1 snd/DA 22.76 A 1 0 351 rcv 22.77 D 601 50 1 snd/DA 22.77 A 1 0 351 rcv 22.78 D 651 50 1 snd/DA 22.78 A 1 0 351 rcv 22.78 D 351 50 1 snd 22.78 A 1 0 551 rcv 22.78 D 751 50 1 snd/DA 22.78 A 1 0 551 rcv 22.78 D 801 50 1 (SKIP LOG UNTIL END)	rcv 3514.48 D 1605151 50 1 snd/DA 3514.48 A 1 0 1604801 rcv 3514.49 D 1605201 50 1 snd/DA 3514.5 A 1 0 1604801 rcv 3514.5 D 1605251 50 1 snd/DA 3514.51 A 1 0 1604801 rcv 3514.54 D 1604801 50 1 snd 3514.55 A 1 0 1605301 rcv 3514.9 D 1605351 50 1 snd/DA 3514.91 A 1 0 1605301 rcv 3514.91 D 1605401 50 1 snd/DA 3514.91 A 1 0 1605301 rcv 3514.91 D 1605401 50 1 snd/DA 3514.91 A 1 0 1605301 rcv 3514.93 D 1605451 50 1 snd/DA 3514.93 A 1 0 1605301 rcv 3514.94 D 1605501 50 1 snd/DA 3514.95 A 1 0 1605301 rcv 3514.98 D 1605301 50 1 snd 3514.98 A 1 0 1605551 rcv 3516.34 D 1605551 35 1 snd 3516.35 A 1 0 1605586 rcv 3516.41 F 1605586 0 1 snd 3516.41 A 1 0 1605587 snd 3516.41 F 1 0 1605587 rcv 3516.41 A 1605587 0 2 ===== Amount of data received (bytes) 1852585 Total Segments Received 37056 Data segments received 37052 Data segments with Bit Errors 3329 Duplicate data segments received 4940 Duplicate ACKs sent 26408 =====	snd 0.0 S 0 0 0 rcv 0.0 SA 0 0 1 snd 0.0 A 1 0 1 snd/corr 14.72 D 1 50 1 snd 14.72 D 51 50 1 snd 14.74 D 101 50 1 rcv 14.73 A 1 0 1 snd 14.75 D 151 50 1 snd 14.77 D 201 50 1 rcv 14.75 A 1 0 1 snd/dup 14.78 D 251 50 1 snd 14.78 D 301 50 1 rcv 14.78 A 1 0 1 snd/corr 14.8 D 351 50 1 snd 14.81 D 401 50 1 rcv 14.8 A 1 0 1 snd 14.83 D 451 50 1 rcv 14.83 A 1 0 1 rcv 14.87 A 1 0 1 rcv 14.95 A 1 0 1 rcv 15.03 A 1 0 1 rcv 15.1 A 1 0 1 snd/RXT 15.72 D 1 50 1 rcv 15.73 A 1 0 351 snd 15.73 D 501 50 1 rcv/DA 15.73 A 1 0 351 snd/corr 15.73 D 551 50 1 snd 15.75 D 601 50 1 rcv/DA 15.75 A 1 0 351 snd 15.75 D 651 50 1 snd/RXT 15.75 D 351 50 1 snd 15.75 D 751 50 1 rcv/DA 15.75 A 1 0 351 snd/dup 15.75 D 801 50 1 rcv 15.75 A 1 0 551 (SKIP LOG UNTIL END)	snd/RXT 3506.24 D 1604601 50 1 rcv 3506.25 A 1 0 1604651 snd 3506.34 D 1605101 50 1 rcv/DA 3506.35 A 1 0 1604651 snd/RXT 3507.33 D 1604651 50 1 rcv 3507.34 A 1 0 1604801 snd/rord 3507.44 D 1604601 50 1 snd/dup 3507.43 D 1605151 50 1 rcv/DA 3507.44 A 1 0 1604801 snd 3507.47 D 1605201 50 1 snd 3507.48 D 1605251 50 1 rcv/DA 3507.47 A 1 0 1604801 snd/RXT 3507.51 D 1604801 50 1 rcv/DA 3507.55 A 1 0 1604801 rcv/DA 3507.62 A 1 0 1604801 rcv/DA 3507.7 A 1 0 1604801 rcv 3507.78 A 1 0 1605301 snd/drop 3507.87 D 1605301 50 1 snd 3507.87 D 1605351 50 1 snd/dup 3507.88 D 1605401 50 1 rcv/DA 3507.88 A 1 0 1605301 snd 3507.9 D 1605451 50 1 snd 3507.92 D 1605501 50 1 rcv/DA 3507.91 A 1 0 1605301 snd/RXT 3507.95 D 1605301 50 1 rcv/DA 3507.99 A 1 0 1605301 rcv/DA 3508.07 A 1 0 1605301 rcv/DA 3508.14 A 1 0 1605301 rcv 3508.22 A 1 0 1605551 snd/RXT 3509.3 D 1605551 35 1 rcv 3509.33 A 1 0 1605586 snd 3509.38 F 1605586 0 1 rcv 3509.38 A 1 0 1605587 rcv 3509.38 F 1 0 1605587 snd 3509.38 A 1605587 0 2 ===== Size of the file (in Bytes) 1605585 Segments transmitted (including drop & RXT) 40857 Number of Segments handled by PLD 40853 Number of Segments dropped 4152 Number of Segments Corrupted 3329 Number of Segments Re-ordered 2261 Number of Segments Duplicated 3684 Number of Segments Delayed 0 Number of Retransmissions due to TIMEOUT 2166 Number of FAST RETRANSMISSION 6575 Number of DUP ACKS received 26399 =====

Table 10: Experiment 3 Receiver Log File

Table 11: Experiment 3 Sender Log File

Event	Percentage of Total Sent
pDrop	4152/40853 = 10%
pCorrupt	3329/40853 = 8%
pDuplicate	3684/40853 = 9%
pOrder	2261/40853 = 5%

Table 12: Experiment 3

References

16 bit IP Hash Algorithm

<https://tools.ietf.org/html/rfc1071>

How to terminate python threads

<https://www.g-loaded.eu/2016/11/24/how-to-terminate-running-python-threads-using-signals/>

Kurose, J. and Ross, K. (2013). Computer networking : A Top-down Approach. 6th Edition. Boston: Pearson.