

1.

Implementation

I made extensive use of ADTs to reduce the complexity of the task and to ensure that changes to one part of the program would not have rippling effects on other parts of the program. The ADTs that were used were:

- **SenderSTP** – contains the main logic for STP for the sender. Receives data from the ‘application layer’. Handles events such as ACKs.
- **SenderWindow** – buffers not-yet-acknowledged segments. Keeps track of *sendBase*. Slides across in response to new ACKs.
- **SenderPLD** – applies errors to to-be-transmitted segments.
- **SenderLogger/ReceiverLogger** – Log events and maintain statistics for the sender/receiver.
- **SenderSocket/ReceiverSocket** – Send and receive STP segments via UDP.
- **Timer** – Runs the timer for the sender. Takes RTT samples. Updates estimated RTT, deviation RTT, and RTO.
- **ReceiverSTP** – contains the main logic for STP for the receiver. Buffers out-of-order segments. Keeps track of *recvBase*.
- **Segment** – encapsulates the creation and parsing of segments.
- **Queue** – enables segments to flow between threads in a consistent order.

One major concern I had was that the receiver could end up missing segments from the sender if it spent too long processing the previous segments, and similarly, the sender could end up missing an ACK if it spent too long processing the previous ACK. Threading was the solution to this problem. At the sender, I created two threads that were solely responsible for transmitting and receiving segments, and additional threads for sending segments through the PLD module, handling ACKs, and running the timer. At the receiver, I created two additional threads: one for receiving data, and one for handling data, including transmitting ACKs. Segments flowed between threads via queues, and mutexes were used to ensure that no piece of data was accessed by two threads at the same time. A diagram of the design is included in Appendix A.

For error detection, I calculated a simple parity bit checksum instead of the Internet checksum used in UDP and TCP. This approach was valid, as we were only required to flip a single bit of data to corrupt our segments.

While testing, I found that even with all of the error probabilities set to zero, timeouts would occasionally occur. This is because the sender and receiver were being run on the same machine, which caused both estimatedRTT and devRTT to fall to extremely low values. To resolve this issue, I imposed a minimum RTO of 0.1 seconds, which is similar to RFC6298’s guideline to use a minimum RTO of 1 second.

Features

I was able to implement all of the required features, which include:

- Three-way handshake for connection establishment
- Four-segment connection termination
- Timer for timeout operation, round-trip-time estimation and RTO estimation

- Duplicate detection
- In-order delivery
- Error detection
- Fast retransmit
- All features of the PLD module (drop, duplicate, corrupt, reorder, delay)
- Pipelining, and ensuring *LastByteSent* – *LastByteAcked* \leq *MWS* at all times

2.

My STP header (each row is 4 bytes):

Sequence number	
Acknowledgement number	
Window size	
Data length	
Flags	Checksum

The STP header is 20 bytes in size and includes these fields:

- **Sequence number** – the same as in TCP. It is the byte stream number of the first byte in the segment's data.
- **Acknowledgement number** – also the same as in TCP. It is the sequence number just beyond the highest sequence number received in order from the other side.
- **Window size** – used by the sender during connection establishment to let the receiver know what the MWS is. The receiver uses this value to ensure enough buffer space is allocated.
- **Data length** – indicates the number of bytes of data in the segment.
- **Flags** – used to indicate whether the segment is an SYN, FIN, ACK or a combination.
- **Checksum** – a simple parity bit checksum.

3.

Design trade-offs

Binary Format Header

A binary format header is harder to read and debug compared to a text-based header, but is easier to parse and less verbose, resulting in less overhead. In this assignment, I took advantage of the ability of a C struct to be treated as simply a sequence of bytes, and used a binary format header. Accessing a particular header field was as simple as accessing a field of the struct, and no additional work was needed when sending the data. Furthermore, with a proper ADT, reading and debugging the header was extremely simple.

Sample RTT calculation

By default, TCP samples RTT approximately once per RTT. My program also does this. Another option that was discussed on the forums was to sample the RTT for every segment that was not retransmitted, by keeping track of when each segment is transmitted, and then calculating the time difference after an ACK for it is received. The latter approach results in a more accurate estimate of the RTT, but is more complex and requires more processing. For simplicity, in this assignment I opted to sample RTT approximately once per whole window.

Threading

The more threads a program uses, the more complex it is, and the more care is needed to properly synchronise the threads. However, I believe the way I have used threads (where each thread is extremely localised and has one purpose) makes my implementation easy to understand.

Possible improvements

Reducing Header Size

Even though the header uses a binary format, and thus already uses less space than a text-based header, it could still have been reduced to a mere 12 bytes. The data length field is not needed, as it can be deduced from the number of bytes received and the header size (which is fixed). (Note that the TCP header does not have a field for data length.) As the checksum is just a parity bit, only one bit is required, but we will leave it as 2 bytes, as it may be desirable to upgrade the checksum to use the TCP Internet checksum instead. Also, there are only three flags (SYN, ACK, FIN), so the flags field need only be one byte in length. As the window size field is only used in the very first SYN segment from the sender to the receiver, and the acknowledgement number field is NOT used in this first segment, we could combine the two fields and give it a dual purpose: if the segment is a SYN segment, the field is used to indicate the window size – otherwise, it is used for the acknowledgement number. The reduced header would look like this:

Sequence number		
Window size/Acknowledgement number		
Flags	Unused	Checksum
Data		

Internet Checksum

For this assignment, we corrupted a segment by simply flipping one bit of data, so using a parity bit for the checksum was sufficient. However, if we were to use STP in the real world, this would only allow us to detect an odd number of bit errors. Using the Internet checksum would allow us to detect, with a high probability, an arbitrary number of bit errors.

Possible extensions

Doubling the Timeout

Although this feature was excluded for the assignment, we could make STP behave more similarly to TCP by doubling the RTO if a timeout occurs. This would be very simple to implement in code.

Delayed ACKs

Another possible feature that would make STP behave more like TCP is to use delayed ACKs. This would require more work, as it would require a timer at the receiver, which we do not currently have.

4.

Borrowed Code

To set up the sockets, I used bits of code from the simple echo client/server on this website: <https://www.binarytides.com/programming-udp-sockets-c-linux/>

5. (a)

pDrop = 0.1

This is the order of sequence numbers of incoming segments observed at the receiver:

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

From the sequence, it is clear that segment 101 was dropped. The arrival of segments 201, 301, and 401 caused the receiver to reply with three duplicate ACKs (101), which in turn triggered fast retransmission of segment 101. The next segments to be dropped were segments 2501 and 2801. The arrival of segments 2601, 2701, and 2901 caused the receiver to reply with three more duplicate ACKs (2501), which triggered fast retransmission of segment 2501. The receiver then replied with ACK 2801. The sender then sent segment 3001, the last segment in its window, which caused the receiver to reply with a duplicate ACK of 2801, as segment 2801 had not yet arrived. Unfortunately, as the sender had no more segments to transmit (its window had slid to [2801, 2901, 3001, ---, ---]), it could not receive enough duplicate ACKs to trigger fast retransmission of segment 2801. Thus, the sender had to wait until the timeout to retransmit segment 2801. After receiving segment 2801, the transfer was complete, and connection teardown followed.

pDrop = 0.3

This is the order of sequence numbers of incoming segments observed at the receiver:

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

As in the above experiment, segment 101 was dropped. This time, it appears that segment 301 was also dropped. The arrival of segments 201, 401, and 501 caused the receiver to reply with three duplicate ACKs (101), which in turn triggered fast retransmission of segment 101. Unfortunately, this retransmission was also dropped (see the log), which forced the sender to wait until the timeout to retransmit segment 101 again. After the arrival of segment 101, the receiver replied with ACK 301, acknowledging segments 101 and 201. This caused the sender's window to slide to [301, 401, 501, 601, 701]. The sender then successfully sent segments 601 and 701. These caused the receiver to reply with two duplicate ACKs (301). Unfortunately, this was one short of the number of duplicate ACKs required for a fast retransmission, so the sender was once again forced to wait until the timeout to retransmit segment 301. From looking at the order of segments arriving at the receiver, it appears that these segments were also dropped: 801, 1101, 1301, 1401, 1601, 1801, 1901, 2401.

The full logs for both experiments are in Appendix B.

Note:

While running experiments for (b) and (c), I observed massive increases in the RTO which meant that it was impractical to continue running them in that state. An inflation in the RTO occurs when a segment (say, X) which is being used to sample the RTT is sent successfully (and is therefore NOT retransmitted), but the acknowledgement for it is not received until much later, because the receiver is still waiting for other segments with a sequence number smaller than X, and those segments have been dropped several times. This causes estimatedRTT AND devRTT and therefore, the RTO, to increase drastically. (For more details, see <https://webcms3.cse.unsw.edu.au/COMP3331/18s2/forums/2709299> - includes staff response.)

5. (b)

To reduce the idle time spent waiting for the timeout and to expedite the transfer, I imposed a maximum RTO of 1 second after the RTO increased beyond 10 seconds. These were the results:

gamma	Number of STP packets transmitted	Transfer time
2	12415	6402 seconds = 106 minutes, 42 seconds
4	12384	6417 seconds = 106 minutes, 57 seconds
6	12389	6352 seconds = 105 minutes, 52 seconds

The first and last 20 log entries and the log summaries for these experiments can be found in Appendix C.

Unfortunately, imposition of a maximum RTO of 1 second obscured the effect of gamma and caused the results to become very homogeneous, and thus not very interesting. I therefore ran the experiments without imposing a maximum RTO for 20 minutes each for each value of gamma, and observed the RTOs. I predicted that the RTOs after 20 minutes would be higher for higher values of gamma. This is what I found:

gamma	RTO after first RTO update
2	9.787 seconds
4	24.075 seconds
6	44.550 seconds

The results were as I expected.

5. (c)

The file was successfully transferred. With an imposed maximum RTO of 1 second, as mentioned above, the file transfer took 3338 seconds = 55 minutes and 38 seconds. The first and last 20 log entries and the log summaries for this experiment can be found in Appendix D.

I also ran the experiment without imposing a maximum RTO for 20 minutes, and observed these RTOs: 1.5s (initial), 7.74s, 39.33s, 51.80s, 55.04s.

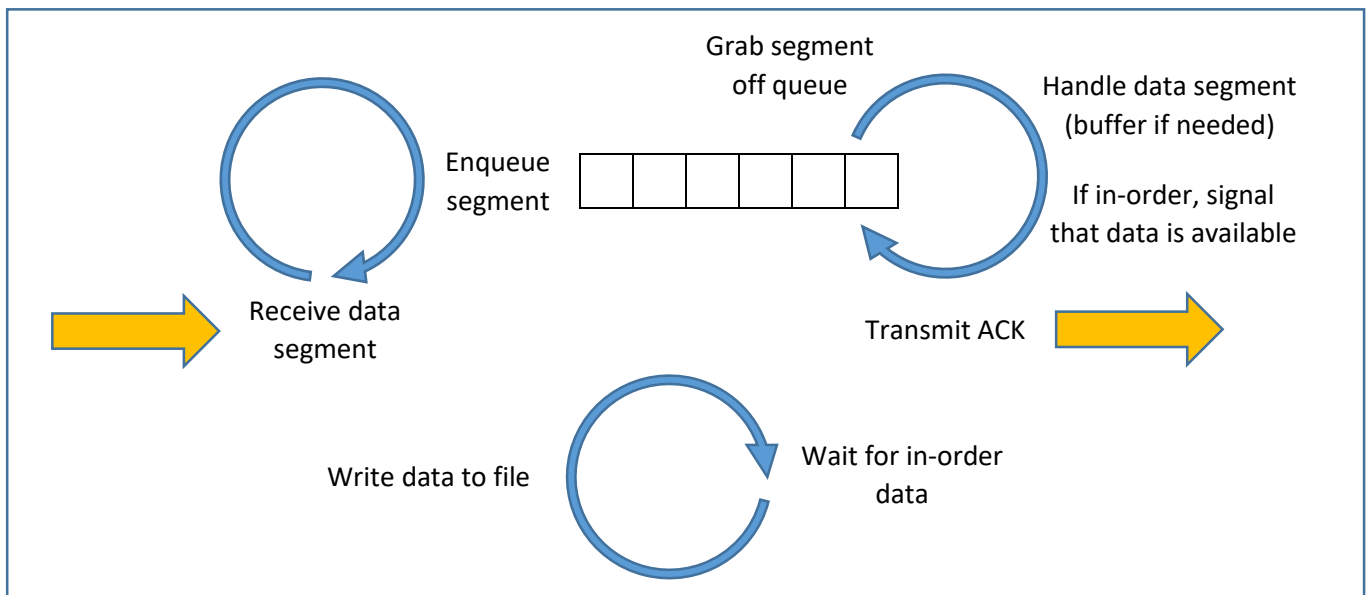
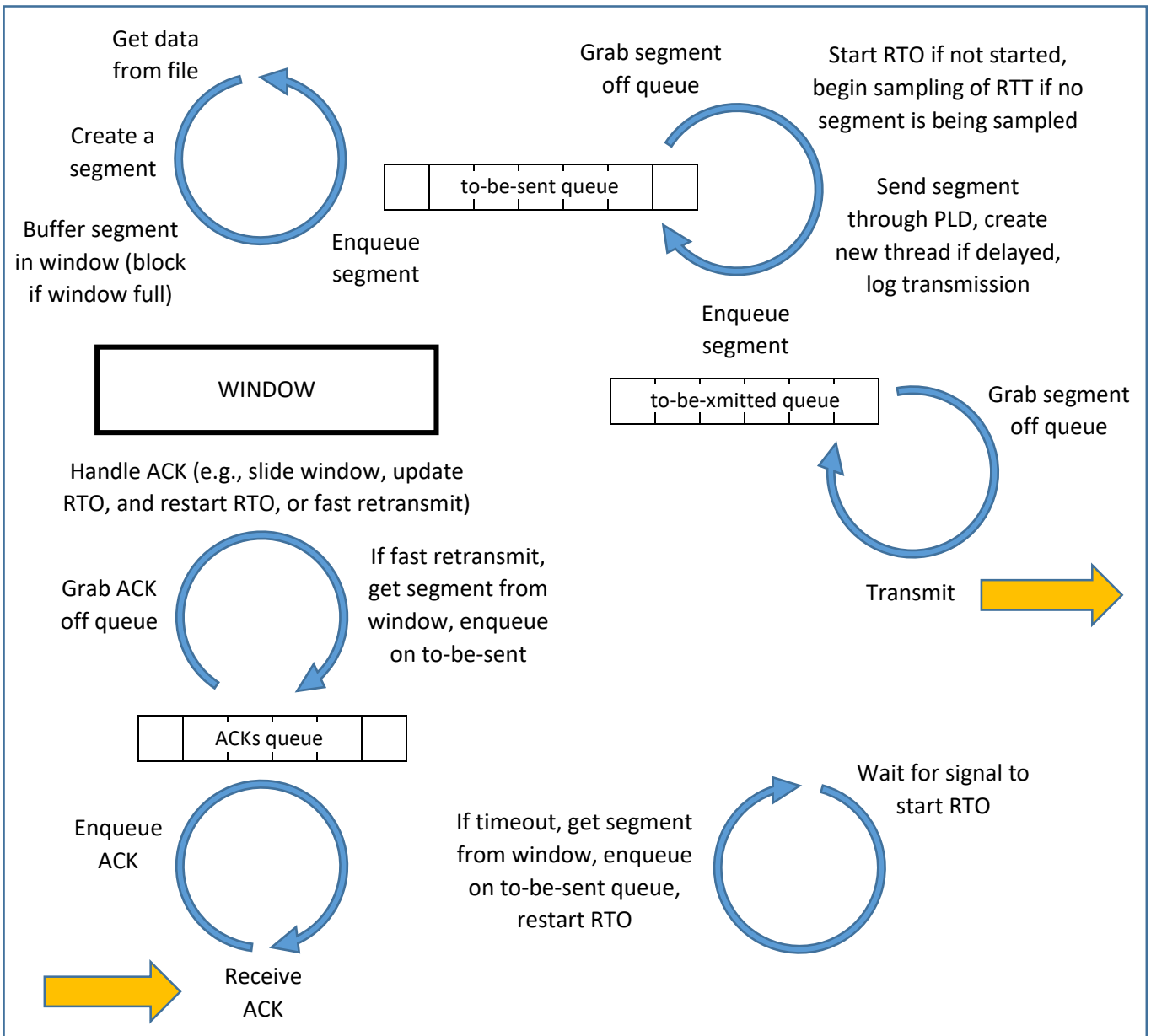
By examining the sender log file for the latter experiment, it is clear that pDrop and pCorrupt are the most critical contributing factors in the overall transfer time. Drops and corruptions are functionally identical, as they both cause a segment to be dropped, and they do NOT cause

duplicate ACKs to be generated, which would help in triggering fast retransmits. Drops and corruptions are also the cause of the scenario I described above. I have included an example of this scenario occurring in Appendix E. Observe that segment 10701 is successfully sent at time 38.28283, but an ACK for it is not received until time 81.27382, because between these two times, the receiver is still waiting for segments with sequence numbers smaller than 10701. Those segments are corrupted and dropped (see times 36.15121, 36.17523, 36.17524, 36.17525, 38.30330, 42.59521, 50.29659). If segment 10701 is being used to sample the RTT (it can, because it is never retransmitted), the sampleRTT would be ~43 seconds, which is very long indeed! This would drastically inflate the estimatedRTT and devRTT, causing the RTO to increase as well.

Ultimately, pDrop would be the most critical contributing factor over pCorrupt, due to probability. Since pDrop is checked first in the PLD module, there is a 10% chance that a segment will be dropped (given a pDrop value of 0.1). Out of the segments that were not dropped (90%), there is a 10% chance that they will be corrupted (given a pCorrupt value of 0.1) ($90\% \times 10\% = 9\%$). So there is a slightly higher chance that a segment will be dropped rather than corrupted.

Appendix

Appendix A. Diagram of implementation



Appendix B. Logs for Question 5 (a), with no max RTO imposed

pDrop = 0.1

Sender_log.txt

event	time	type	seq	data	ack
snd	0.00001	S	0	0	0
rcv	0.00029	SA	0	0	1
snd	0.00031	A	1	0	1
snd	0.00114	D	1	100	1
snd/drop	0.00130	D	101	100	1
snd	0.00131	D	201	100	1
snd	0.00131	D	301	100	1
snd	0.00132	D	401	100	1
rcv	0.01812	A	1	0	101
rcv/DA	0.01813	A	1	0	101
rcv/DA	0.01814	A	1	0	101
rcv/DA	0.01814	A	1	0	101
snd/RXT	0.01815	D	101	100	1
rcv	0.01836	A	1	0	501
snd	0.02994	D	501	100	1
snd	0.02995	D	601	100	1
snd	0.02995	D	701	100	1
snd	0.02995	D	801	100	1
snd	0.02995	D	901	100	1
rcv	0.04211	A	1	0	601
rcv	0.04212	A	1	0	701
rcv	0.04212	A	1	0	801
snd	0.04213	D	1001	100	1
rcv	0.04214	A	1	0	901
snd	0.04215	D	1101	100	1
snd	0.04216	D	1201	100	1
snd	0.04216	D	1301	100	1
rcv	0.04218	A	1	0	1001
snd	0.04224	D	1401	100	1
rcv	0.04228	A	1	0	1101
rcv	0.04229	A	1	0	1201
rcv	0.04232	A	1	0	1301
rcv	0.04244	A	1	0	1401
rcv	0.04246	A	1	0	1501
snd	0.04254	D	1501	100	1
snd	0.04254	D	1601	100	1
snd	0.04254	D	1701	100	1
snd	0.04262	D	1801	100	1
snd	0.04262	D	1901	100	1
rcv	0.04266	A	1	0	1601
rcv	0.04269	A	1	0	1701
rcv	0.04271	A	1	0	1801
rcv	0.04273	A	1	0	1901
rcv	0.04275	A	1	0	2001
snd	0.06211	D	2001	100	1
snd	0.06211	D	2101	100	1
snd	0.06211	D	2201	100	1
snd	0.06211	D	2301	100	1
snd	0.06211	D	2401	100	1
rcv	0.07411	A	1	0	2101
rcv	0.07411	A	1	0	2201
rcv	0.07412	A	1	0	2301
rcv	0.07412	A	1	0	2401
rcv	0.07413	A	1	0	2501
snd/drop	0.08611	D	2501	100	1
snd	0.08612	D	2601	100	1
snd	0.08612	D	2701	100	1
snd/drop	0.08612	D	2801	100	1
snd	0.08612	D	2901	100	1

rcv/DA	0.09810	A	1	0	2501
rcv/DA	0.09811	A	1	0	2501
rcv/DA	0.09811	A	1	0	2501
snd/RXT	0.09812	D	2501	100	1
rcv	0.09830	A	1	0	2801
snd	0.11010	D	3001	28	1
rcv/DA	0.12210	A	1	0	2801
snd/RXT	1.54741	D	2801	100	1
rcv	1.57410	A	1	0	3029
snd	1.57667	F	3029	0	1
rcv	1.58610	A	1	0	3030
rcv	1.58612	F	1	0	3030
snd	1.58784	A	3030	0	2

```

=====
Size of the file (in bytes)                               3028
Segments transmitted (including drop & RXT)              38
Number of segments handled by PLD                        34
Number of segments dropped                               3
Number of segments duplicated                             0
Number of segments corrupted                             0
Number of segments reordered                             0
Number of segments delayed                               0
Number of retransmissions due to TIMEOUT                 1
Number of FAST RETRANSMISSIONs                           2
Number of DUP ACKs received                             7
=====

```

Receiver_log.txt

event	time	type	seq	data	ack
rcv	0.58385	S	0	0	0
snd	0.58388	SA	0	0	1
rcv	0.58411	A	1	0	1
rcv	0.58991	D	1	100	1
snd	0.58998	A	1	0	101
rcv	0.59004	D	201	100	1
snd/DA	0.59005	A	1	0	101
rcv	0.59011	D	301	100	1
snd/DA	0.59014	A	1	0	101
rcv	0.59019	D	401	100	1
snd/DA	0.59023	A	1	0	101
rcv	0.60194	D	101	100	1
snd	0.60205	A	1	0	501
rcv	0.61378	D	501	100	1
snd	0.61383	A	1	0	601
rcv	0.61385	D	601	100	1
snd	0.61386	A	1	0	701
rcv	0.61388	D	701	100	1
snd	0.61389	A	1	0	801
rcv	0.61390	D	801	100	1
snd	0.61391	A	1	0	901
rcv	0.61393	D	901	100	1
snd	0.61394	A	1	0	1001
rcv	0.62587	D	1001	100	1
snd	0.62593	A	1	0	1101
rcv	0.62595	D	1101	100	1
snd	0.62596	A	1	0	1201
rcv	0.62600	D	1201	100	1
snd	0.62601	A	1	0	1301
rcv	0.62609	D	1301	100	1
snd	0.62610	A	1	0	1401
rcv	0.62614	D	1401	100	1
snd	0.62615	A	1	0	1501
rcv	0.62630	D	1501	100	1
snd	0.62631	A	1	0	1601

rcv	0.62637	D	1601	100	1
snd	0.62638	A	1	0	1701
rcv	0.62639	D	1701	100	1
snd	0.62640	A	1	0	1801
rcv	0.62641	D	1801	100	1
snd	0.62642	A	1	0	1901
rcv	0.62643	D	1901	100	1
snd	0.62644	A	1	0	2001
rcv	0.64587	D	2001	100	1
snd	0.64592	A	1	0	2101
rcv	0.64594	D	2101	100	1
snd	0.64595	A	1	0	2201
rcv	0.64597	D	2201	100	1
snd	0.64598	A	1	0	2301
rcv	0.64604	D	2301	100	1
snd	0.64605	A	1	0	2401
rcv	0.64611	D	2401	100	1
snd	0.64612	A	1	0	2501
rcv	0.66991	D	2601	100	1
snd/DA	0.66991	A	1	0	2501
rcv	0.66996	D	2701	100	1
snd/DA	0.66996	A	1	0	2501
rcv	0.67002	D	2901	100	1
snd/DA	0.67003	A	1	0	2501
rcv	0.68193	D	2501	100	1
snd	0.68198	A	1	0	2801
rcv	0.69383	D	3001	28	1
snd/DA	0.69384	A	1	0	2801
rcv	2.14194	D	2801	100	1
snd	2.14199	A	1	0	3029
rcv	2.16045	F	3029	0	1
snd	2.16050	A	1	0	3030
snd	2.16284	F	1	0	3030
rcv	2.17157	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                  7
=====

```

pDrop = 0.3

Sender_log.txt

event	time	type	seq	data	ack
snd	0.00001	S	0	0	0
rcv	0.00028	SA	0	0	1
snd	0.00030	A	1	0	1
snd	0.00135	D	1	100	1
snd/drop	0.00151	D	101	100	1
snd	0.00152	D	201	100	1
snd/drop	0.00167	D	301	100	1
snd	0.00167	D	401	100	1
rcv	0.01585	A	1	0	101
rcv/DA	0.01586	A	1	0	101
rcv/DA	0.01587	A	1	0	101
snd	0.02233	D	501	100	1
rcv/DA	0.03187	A	1	0	101
snd/RXT/drop	0.03188	D	101	100	1
snd/RXT	1.62989	D	101	100	1
rcv	1.64183	A	1	0	301
snd	1.64984	D	601	100	1

snd	1.64985	D	701	100	1
rcv/DA	1.65719	A	1	0	301
rcv/DA	1.65720	A	1	0	301
snd/RXT	3.25585	D	301	100	1
rcv	3.25624	A	1	0	801
snd/drop	3.26388	D	801	100	1
snd	3.26388	D	901	100	1
snd	3.26389	D	1001	100	1
rcv/DA	3.26409	A	1	0	801
rcv/DA	3.26414	A	1	0	801
snd/drop	3.27187	D	1101	100	1
snd	3.27188	D	1201	100	1
rcv/DA	3.27203	A	1	0	801
snd/RXT	3.27207	D	801	100	1
rcv	3.27235	A	1	0	1101
snd/drop	3.29588	D	1301	100	1
snd/drop	3.29589	D	1401	100	1
snd	3.29589	D	1501	100	1
rcv/DA	3.29609	A	1	0	1101
snd/RXT	7.38724	D	1101	100	1
rcv	7.38754	A	1	0	1301
snd/drop	7.39987	D	1601	100	1
snd	7.39988	D	1701	100	1
rcv/DA	7.40005	A	1	0	1301
snd/RXT/drop	11.50250	D	1301	100	1
snd/RXT/drop	15.61734	D	1301	100	1
snd/RXT	19.73223	D	1301	100	1
rcv	19.75590	A	1	0	1401
snd/drop	19.75596	D	1801	100	1
snd/RXT	23.87080	D	1401	100	1
rcv	23.89597	A	1	0	1601
snd/drop	23.89747	D	1901	100	1
snd	23.89747	D	2001	100	1
rcv/DA	23.89758	A	1	0	1601
snd/RXT	28.01090	D	1601	100	1
rcv	28.01117	A	1	0	1801
snd	28.01988	D	2101	100	1
snd	28.01988	D	2201	100	1
rcv/DA	28.03064	A	1	0	1801
rcv/DA	28.03070	A	1	0	1801
snd/RXT	32.12609	D	1801	100	1
rcv	32.12633	A	1	0	1901
snd	32.13592	D	2301	100	1
rcv/DA	32.14805	A	1	0	1901
snd/RXT	36.24127	D	1901	100	1
rcv	36.25189	A	1	0	2401
snd/drop	36.26388	D	2401	100	1
snd	36.26388	D	2501	100	1
snd	36.26388	D	2601	100	1
snd	36.26388	D	2701	100	1
snd	36.26388	D	2801	100	1
rcv/DA	36.27588	A	1	0	2401
rcv/DA	36.27588	A	1	0	2401
rcv/DA	36.27589	A	1	0	2401
rcv/DA	36.27589	A	1	0	2401
snd/RXT	36.28787	D	2401	100	1
rcv	36.29989	A	1	0	2901
snd	36.30788	D	2901	100	1
snd/drop	36.30789	D	3001	28	1
rcv	36.31990	A	1	0	3001
snd/RXT	42.47347	D	3001	28	1
rcv	42.47988	A	1	0	3029
snd	42.48174	F	3029	0	1
rcv	42.48371	A	1	0	3030
rcv	42.48528	F	1	0	3030
snd	42.48705	A	3030	0	2

```

=====
Size of the file (in bytes)                3028
Segments transmitted (including drop & RXT) 49
Number of segments handled by PLD          45
Number of segments dropped                 14
Number of segments duplicated              0
Number of segments corrupted               0
Number of segments reordered               0
Number of segments delayed                 0
Number of retransmissions due to TIMEOUT  11
Number of FAST RETRANSMISSIONs            3
Number of DUP ACKs received                18
=====

```

Receiver_log.txt

event	time	type	seq	data	ack
rcv	0.12844	S	0	0	0
snd	0.12846	SA	0	0	1
rcv	0.12870	A	1	0	1
rcv	0.12996	D	1	100	1
snd	0.13010	A	1	0	101
rcv	0.13019	D	201	100	1
snd/DA	0.13033	A	1	0	101
rcv	0.13052	D	401	100	1
snd/DA	0.13065	A	1	0	101
rcv	0.15094	D	501	100	1
snd/DA	0.15098	A	1	0	101
rcv	1.75832	D	101	100	1
snd	1.75840	A	1	0	301
rcv	1.77820	D	601	100	1
snd/DA	1.77824	A	1	0	301
rcv	1.77830	D	701	100	1
snd/DA	1.77834	A	1	0	301
rcv	3.38428	D	301	100	1
snd	3.38440	A	1	0	801
rcv	3.39227	D	901	100	1
snd/DA	3.39231	A	1	0	801
rcv	3.39233	D	1001	100	1
snd/DA	3.39234	A	1	0	801
rcv	3.40022	D	1201	100	1
snd/DA	3.40026	A	1	0	801
rcv	3.40046	D	801	100	1
snd	3.40056	A	1	0	1101
rcv	3.42427	D	1501	100	1
snd/DA	3.42431	A	1	0	1101
rcv	7.51561	D	1101	100	1
snd	7.51572	A	1	0	1301
rcv	7.52823	D	1701	100	1
snd/DA	7.52827	A	1	0	1301
rcv	19.88021	D	1301	100	1
snd	19.88026	A	1	0	1401
rcv	24.01618	D	1401	100	1
snd	24.01619	A	1	0	1601
rcv	24.02580	D	2001	100	1
snd/DA	24.02580	A	1	0	1601
rcv	28.13932	D	1601	100	1
snd	28.13943	A	1	0	1801
rcv	28.14827	D	2101	100	1
snd/DA	28.14831	A	1	0	1801
rcv	28.14836	D	2201	100	1
snd/DA	28.14841	A	1	0	1801
rcv	32.25446	D	1801	100	1
snd	32.25458	A	1	0	1901
rcv	32.27622	D	2301	100	1
snd/DA	32.27626	A	1	0	1901

rcv	36.36968	D	1901	100	1
snd	36.36979	A	1	0	2401
rcv	36.39223	D	2501	100	1
snd/DA	36.39223	A	1	0	2401
rcv	36.39226	D	2601	100	1
snd/DA	36.39227	A	1	0	2401
rcv	36.39232	D	2701	100	1
snd/DA	36.39236	A	1	0	2401
rcv	36.39240	D	2801	100	1
snd/DA	36.39244	A	1	0	2401
rcv	36.41630	D	2401	100	1
snd	36.41643	A	1	0	2901
rcv	36.43625	D	2901	100	1
snd	36.43636	A	1	0	3001
rcv	42.60185	D	3001	28	1
snd	42.60196	A	1	0	3029
rcv	42.61010	F	3029	0	1
snd	42.61014	A	1	0	3030
snd	42.61243	F	1	0	3030
rcv	42.61539	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                  18
=====

```

Appendix C. Logs for Question 5 (b), with a max RTO of 1 second imposed

i. $\gamma = 2$

Sender_log.txt (first and last 20 entries, not including connection establishment/teardown)

event	time	type	seq	data	ack
snd	0.00001	S	0	0	0
rcv	0.00011	SA	0	0	1
snd	0.00012	A	1	0	1
snd/drop	0.01678	D	1	50	1
snd/drop	0.01679	D	51	50	1
snd/drop	0.01680	D	101	50	1
snd	0.01680	D	151	50	1
snd/drop	0.01680	D	201	50	1
snd/drop	0.01681	D	251	50	1
snd/drop	0.01681	D	301	50	1
snd/drop	0.01682	D	351	50	1
snd	0.01682	D	401	50	1
snd	0.01682	D	451	50	1
rcv/DA	0.07676	A	1	0	1
rcv/DA	0.07676	A	1	0	1
rcv/DA	0.07676	A	1	0	1
snd/RXT/drop	0.08873	D	1	50	1
snd/RXT/drop	1.02075	D	1	50	1
snd/RXT/drop	2.02073	D	1	50	1
snd/RXT/drop	3.04474	D	1	50	1
snd/RXT/drop	4.03273	D	1	50	1
snd/RXT/drop	5.02075	D	1	50	1
snd/RXT/drop	6.02075	D	1	50	1
...					

snd/RXT/drop	6389.94473	D	307651	50	1
snd/RXT/drop	6390.93290	D	307651	50	1
snd/RXT	6391.93296	D	307651	50	1
rcv	6391.94477	A	1	0	307701
snd/drop	6391.96873	D	308151	50	1
snd/RXT	6392.96873	D	307701	50	1
rcv	6393.01676	A	1	0	307801
snd/drop	6393.04073	D	308201	3	1
snd/RXT	6394.04073	D	307801	50	1
rcv	6394.05688	A	1	0	308051
snd/RXT	6395.08073	D	308051	50	1
rcv	6395.13674	A	1	0	308101
snd/RXT/drop	6396.16073	D	308101	50	1
snd/RXT	6397.14873	D	308101	50	1
rcv	6397.20072	A	1	0	308151
snd/RXT/dely	6398.95178	D	308151	50	1
rcv	6398.97273	A	1	0	308201
snd/RXT/drop	6399.97273	D	308201	3	1
snd/RXT	6400.97274	D	308201	3	1
rcv	6401.02077	A	1	0	308204
snd	6401.03473	F	308204	0	1
rcv	6401.04889	A	1	0	308205
rcv	6401.60600	F	1	0	308205
snd	6401.64901	A	308205	0	2

```

=====
Size of the file (in bytes)                308203
Segments transmitted (including drop & RXT) 12415
Number of segments handled by PLD          12411
Number of segments dropped                 6224
Number of segments corrupted                0
Number of segments duplicated               0
Number of segments reordered               0
Number of segments delayed                  1214
Number of retransmissions due to TIMEOUT  5989
Number of FAST RETRANSMISSIONs            257
Number of DUP ACKs received                3116
=====

```

Receiver_log.txt (first and last 20 entries, not including connection establishment/teardown)

event	time	type	seq	data	ack
rcv	1.02796	S	0	0	0
snd	1.02798	SA	0	0	1
rcv	1.02803	A	1	0	1
rcv	1.08778	D	151	50	1
snd/DA	1.08779	A	1	0	1
rcv	1.08789	D	401	50	1
snd/DA	1.08790	A	1	0	1
rcv	1.08799	D	451	50	1
snd/DA	1.08800	A	1	0	1
rcv	9.07086	D	1	50	1
snd	9.07095	A	1	0	51
rcv	14.09886	D	51	50	1
snd	14.09894	A	1	0	101
rcv	18.12529	D	101	50	1
snd	18.12537	A	1	0	201
rcv	18.20464	D	601	50	1
snd/DA	18.20466	A	1	0	201
rcv	19.26924	D	201	50	1
snd	19.26932	A	1	0	251
rcv	19.32068	D	701	50	1
snd/DA	19.32069	A	1	0	251
rcv	24.34688	D	251	50	1
snd	24.34697	A	1	0	301
...					

```

...
rcv          6387.93884      D      307501      50          1
snd          6387.93891      A          1          0      307551
rcv          6388.76707      D      308001      50          1
snd/DA       6388.76707      A          1          0      307551
rcv          6389.93917      D      307551      50          1
snd          6389.93928      A          1          0      307651
rcv          6392.97134      D      307651      50          1
snd          6392.97144      A          1          0      307701
rcv          6394.02864      D      307701      50          1
snd          6394.02873      A          1          0      307801
rcv          6395.08463      D      307801      50          1
snd          6395.08471      A          1          0      308051
rcv          6396.13262      D      308051      50          1
snd          6396.13271      A          1          0      308101
rcv          6398.20463      D      308101      50          1
snd          6398.20472      A          1          0      308151
rcv          6399.99018      D      308151      50          1
snd          6399.99026      A          1          0      308201
rcv          6402.03264      D      308201          3          1
snd          6402.03272      A          1          0      308204
rcv          6402.07664      F      308204          0          1
snd          6402.07674      A          1          0      308205
snd          6402.63387      F          1          0      308205
rcv          6402.67695      A      308205          0          2

```

```

=====
Amount of data received (bytes)      309303
Total segments received              6191
Data segments received               6187
Data segments with bit errors        0
Duplicate data segments received      22
Duplicate ACKs sent                   3116
=====

```

ii. gamma = 4

Sender_log.txt (first and last 20 entries, not including connection establishment/teardown)

```

event      time      type      seq      data      ack
snd         0.00001    S         0         0         0
rcv         0.00012    SA        0         0         1
snd         0.00013    A         1         0         1
snd/drop    0.01014    D         1         50        1
snd/drop    0.01016    D         51        50        1
snd/drop    0.01017    D        101        50        1
snd         0.01017    D        151        50        1
snd/drop    0.01033    D        201        50        1
snd/drop    0.01042    D        251        50        1
snd/drop    0.01042    D        301        50        1
snd/drop    0.01043    D        351        50        1
snd         0.01043    D        401        50        1
snd         0.01043    D        451        50        1
rcv/DA      0.07840    A         1         0         1
rcv/DA      0.07841    A         1         0         1
rcv/DA      0.07842    A         1         0         1
snd/RXT/drop 0.08665    D         1         50        1
snd/RXT/drop 1.53466    D         1         50        1
snd/RXT/drop 3.03483    D         1         50        1
snd/RXT/drop 4.54548    D         1         50        1
snd/RXT/drop 6.04562    D         1         50        1
snd/RXT/drop 7.53509    D         1         50        1
snd/RXT/drop 9.03524    D         1         50        1
...

```

snd/RXT	6403.73531	D	307701	50	1
rcv	6403.76243	A	1	0	307751
snd/drop	6403.77439	D	308201	3	1
snd/RXT/drop	6404.77297	D	307751	50	1
snd/RXT/drop	6405.76259	D	307751	50	1
snd/RXT	6406.76266	D	307751	50	1
rcv	6406.79525	A	1	0	307901
snd/RXT	6407.80582	D	307901	50	1
rcv	6407.87044	A	1	0	307951
snd/RXT/dely	6409.51344	D	307951	50	1
rcv	6409.54253	A	1	0	308051
snd/RXT/drop	6410.55310	D	308051	50	1
snd/RXT/dely	6411.72844	D	308051	50	1
rcv	6411.75842	A	1	0	308101
snd/RXT/drop	6412.78408	D	308101	50	1
snd/RXT/drop	6413.75859	D	308101	50	1
snd/RXT	6414.77839	D	308101	50	1
rcv	6414.79456	A	1	0	308201
snd/RXT/dely	6416.60544	D	308201	3	1
rcv	6416.62147	A	1	0	308204
snd	6416.62360	F	308204	0	1
rcv	6416.62387	A	1	0	308205
rcv	6416.62657	F	1	0	308205
snd	6416.62860	A	308205	0	2

```

=====
Size of the file (in bytes)                308203
Segments transmitted (including drop & RXT) 12384
Number of segments handled by PLD          12380
Number of segments dropped                 6199
Number of segments duplicated              0
Number of segments corrupted               0
Number of segments reordered              0
Number of segments delayed                 1219
Number of retransmissions due to TIMEOUT  5966
Number of FAST RETRANSMISSIONs            249
Number of DUP ACKs received               3092
=====

```

Receiver_log.txt (first and last 20 entries, not including connection establishment/teardown)

event	time	type	seq	data	ack
rcv	0.39868	S	0	0	0
snd	0.39869	SA	0	0	1
rcv	0.39910	A	1	0	1
rcv	0.44503	D	151	50	1
snd/DA	0.44504	A	1	0	1
rcv	0.44517	D	401	50	1
snd/DA	0.44518	A	1	0	1
rcv	0.45425	D	451	50	1
snd/DA	0.45425	A	1	0	1
rcv	12.44455	D	1	50	1
snd	12.47698	A	1	0	51
rcv	20.00098	D	51	50	1
snd	20.00106	A	1	0	101
rcv	26.03204	D	101	50	1
snd	26.04223	A	1	0	201
rcv	26.11298	D	601	50	1
snd/DA	26.11300	A	1	0	201
rcv	27.69698	D	201	50	1
snd	27.69705	A	1	0	251
rcv	27.76100	D	701	50	1
snd/DA	27.76102	A	1	0	251
rcv	35.25300	D	251	50	1
snd	35.25305	A	1	0	301
...					


```

...
rcv          6401.78101      D      308001      50          1
snd/DA       6401.78102      A          1          0      307551
rcv          6403.11301      D      307551      50          1
snd          6403.11308      A          1          0      307701
rcv          6403.14901      D      308151      50          1
snd/DA       6403.14902      A          1          0      307701
rcv          6404.14927      D      307701      50          1
snd          6404.14940      A          1          0      307751
rcv          6407.18498      D      307751      50          1
snd          6407.18520      A          1          0      307901
rcv          6408.25698      D      307901      50          1
snd          6408.25721      A          1          0      307951
rcv          6409.94098      D      307951      50          1
snd          6409.94105      A          1          0      308051
rcv          6412.13703      D      308051      50          1
snd          6412.14723      A          1          0      308101
rcv          6415.18287      D      308101      50          1
snd          6415.19304      A          1          0      308201
rcv          6417.00906      D      308201          3          1
snd          6417.00932      A          1          0      308204
rcv          6417.02223      F      308204          0          1
snd          6417.02244      A          1          0      308205
snd          6417.02507      F          1          0      308205
rcv          6417.02722      A      308205          0          2

```

```

=====
Amount of data received (bytes)      309003
Total segments received              6185
Data segments received               6181
Data segments with bit errors        0
Duplicate data segments received      16
Duplicate ACKs sent                   3092
=====

```

iii. gamma = 6

Sender_log.txt (first and last 20 entries, not including connection establishment/teardown)

```

event      time      type      seq      data      ack
snd         0.00002    S         0         0         0
rcv         0.00033    SA        0         0         1
snd         0.00035    A         1         0         1
snd/drop    0.00216    D         1         50        1
snd/drop    0.00233    D         51        50        1
snd/drop    0.00248    D        101        50        1
snd         0.00249    D        151        50        1
snd/drop    0.00264    D        201        50        1
snd/drop    0.00274    D        251        50        1
snd/drop    0.00275    D        301        50        1
snd/drop    0.00276    D        351        50        1
snd         0.00277    D        401        50        1
snd         0.00277    D        451        50        1
rcv/DA      0.01152    A         1         0         1
rcv/DA      0.01153    A         1         0         1
rcv/DA      0.01153    A         1         0         1
snd/RXT/drop 0.01159    D         1         50        1
snd/RXT/drop 2.00928    D         1         50        1
snd/RXT/drop 4.00934    D         1         50        1
snd/RXT/drop 6.00947    D         1         50        1
snd/RXT/drop 8.00948    D         1         50        1
snd/RXT/drop 10.00955   D         1         50        1
snd/RXT/drop 12.01719   D         1         50        1
...

```

snd/RXT/drop	6340.18519	D	307701	50	1
snd/RXT	6341.17334	D	307701	50	1
rcv	6341.19328	A	1	0	307751
snd	6341.19718	D	308201	3	1
rcv/DA	6341.20510	A	1	0	307751
snd/RXT/dely	6342.54503	D	307751	50	1
rcv	6342.57321	A	1	0	307901
snd/RXT/drop	6343.57330	D	307901	50	1
snd/RXT/drop	6344.57719	D	307901	50	1
snd/RXT	6345.58518	D	307901	50	1
rcv	6345.59719	A	1	0	307951
snd/RXT	6346.60918	D	307951	50	1
rcv	6346.62150	A	1	0	308101
snd/RXT/drop	6347.62155	D	308101	50	1
snd/RXT/drop	6348.63319	D	308101	50	1
snd/RXT/drop	6349.62169	D	308101	50	1
snd/RXT	6350.63318	D	308101	50	1
rcv	6350.63341	A	1	0	308151
snd/RXT	6351.63352	D	308151	50	1
rcv	6351.64919	A	1	0	308204
snd	6351.65089	F	308204	0	1
rcv	6351.65373	A	1	0	308205
rcv	6351.65373	F	1	0	308205
snd	6351.65545	A	308205	0	2

```

=====
Size of the file (in bytes)                308203
Segments transmitted (including drop & RXT) 12389
Number of segments handled by PLD          12385
Number of segments dropped                 6207
Number of segments duplicated              0
Number of segments corrupted               0
Number of segments reordered               0
Number of segments delayed                 1200
Number of retransmissions due to TIMEOUT  5965
Number of FAST RETRANSMISSIONs            255
Number of DUP ACKs received               3115
=====

```

Receiver_log.txt (first and last 20 entries, not including connection establishment/teardown)

event	time	type	seq	data	ack
rcv	0.32961	S	0	0	0
snd	0.32966	SA	0	0	1
rcv	0.32988	A	1	0	1
rcv	0.33259	D	151	50	1
snd/DA	0.33261	A	1	0	1
rcv	0.33308	D	401	50	1
snd/DA	0.33310	A	1	0	1
rcv	0.33336	D	451	50	1
snd/DA	0.33337	A	1	0	1
rcv	16.34103	D	1	50	1
snd	16.34116	A	1	0	51
rcv	26.35508	D	51	50	1
snd	26.35520	A	1	0	101
rcv	34.36700	D	101	50	1
snd	34.36714	A	1	0	201
rcv	34.37862	D	601	50	1
snd/DA	34.37863	A	1	0	201
rcv	36.46082	D	201	50	1
snd	36.46090	A	1	0	251
rcv	36.47070	D	701	50	1
snd/DA	36.47181	A	1	0	251
rcv	46.46133	D	251	50	1
snd	46.47060	A	1	0	301
...					

```

...
rcv          6336.49071      D      308001      50          1
snd/DA       6336.49075      A          1          0      307651
rcv          6336.49081      D      308051      50          1
snd/DA       6336.49085      A          1          0      307651
rcv          6339.49074      D      307651      50          1
snd          6339.49082      A          1          0      307701
rcv          6341.51461      D      307701      50          1
snd          6341.51463      A          1          0      307751
rcv          6341.53441      D      308201      3           1
snd/DA       6341.53445      A          1          0      307751
rcv          6342.88261      D      307751      50          1
snd          6342.88263      A          1          0      307901
rcv          6345.91473      D      307901      50          1
snd          6345.91481      A          1          0      307951
rcv          6346.95069      D      307951      50          1
snd          6346.95077      A          1          0      308101
rcv          6350.96272      D      308101      50          1
snd          6350.96280      A          1          0      308151
rcv          6351.96668      D      308151      50          1
snd          6351.96677      A          1          0      308204
rcv          6351.98039      F      308204      0           1
snd          6351.98043      A          1          0      308205
snd          6351.98298      F          1          0      308205
rcv          6351.98488      A      308205      0           2

```

```

=====
Amount of data received (bytes)      308853
Total segments received              6182
Data segments received              6178
Data segments with bit errors        0
Duplicate data segments received      13
Duplicate ACKs sent                  3115
=====

```

Appendix D. Logs for Question 5 (c), with a max RTO of 1 second imposed

Sender_log.txt (first and last 20 entries, not including connection establishment/teardown)

```

event          time          type      seq      data      ack
snd            0.00001        S          0         0         0
rcv            0.00018        SA         0         0         1
snd            0.00019        A          1         0         1
snd            0.03051        D          1        50         1
snd            0.03052        D         51        50         1
snd            0.03054        D        101        50         1
snd/dup        0.03054        D        101        50         1
snd            0.03054        D        151        50         1
snd/corr       0.03055        D        200        50         1
snd            0.03055        D        251        50         1
snd            0.03056        D        301        50         1
snd            0.03056        D        401        50         1
snd            0.03057        D        451        50         1
rcv            0.07849        A          1         0        51
rcv            0.07852        A          1         0       101
rcv            0.07855        A          1         0       151
rcv/DA        0.07856        A          1         0       151
rcv            0.07858        A          1         0       201
rcv/DA        0.07860        A          1         0       201
rcv/DA        0.07861        A          1         0       201
rcv/DA        0.07862        A          1         0       201
rcv/DA        0.07863        A          1         0       201
snd/RXT/corr   0.10246        D        201        50         1
...

```

rcv	3336.49048	A	1	0	1605251
snd	3336.53846	D	1605301	50	1
snd	3336.53846	D	1605351	50	1
snd	3336.53850	D	1605401	50	1
snd/dup	3336.53850	D	1605401	50	1
snd	3336.53850	D	1605451	50	1
snd	3336.53851	D	1605551	35	1
rcv/DA	3336.56246	A	1	0	1605251
rcv/DA	3336.56246	A	1	0	1605251
rcv/DA	3336.56247	A	1	0	1605251
rcv/DA	3336.56247	A	1	0	1605251
rcv/DA	3336.56247	A	1	0	1605251
rcv/DA	3336.56248	A	1	0	1605251
snd/RXT	3336.57445	D	1605251	50	1
snd/RXT/drop	3336.57446	D	1605251	50	1
rcv	3336.59848	A	1	0	1605501
snd/RXT	3337.59855	D	1605501	50	1
snd/rord	3337.59855	D	1605501	50	1
rcv	3337.61047	A	1	0	1605586
rcv/DA	3337.61047	A	1	0	1605586
snd	3337.62475	F	1605586	0	1
rcv	3337.64057	A	1	0	1605587
rcv	3338.24347	F	1	0	1605587
snd	3338.26863	A	1605587	0	2

```

=====
Size of the file (in bytes)                1605585
Segments transmitted (including drop & RXT) 45683
Number of segments handled by PLD          45679
Number of segments dropped                 4286
Number of segments corrupted               3250
Number of segments duplicated              3790
Number of segments reordered               2420
Number of segments delayed                  0
Number of retransmissions due to TIMEOUT  3113
Number of FAST RETRANSMISSIONs            7555
Number of DUP ACKs received                27956
=====

```

Receiver_log.txt (first and last 20 entries, not including connection establishment/teardown)

event	time	type	seq	data	ack
rcv	0.37099	S	0	0	0
snd	0.37100	SA	0	0	1
rcv	0.37111	A	1	0	1
rcv	0.44134	D	1	50	1
snd	0.44150	A	1	0	51
rcv	0.44160	D	51	50	1
snd	0.44165	A	1	0	101
rcv	0.44174	D	101	50	1
snd	0.44179	A	1	0	151
rcv	0.44189	D	101	50	1
snd/DA	0.44192	A	1	0	151
rcv	0.44198	D	151	50	1
snd	0.44203	A	1	0	201
rcv/corr	0.44212	D	200	50	1
rcv	0.44213	D	251	50	1
snd/DA	0.44213	A	1	0	201
rcv	0.44222	D	301	50	1
snd/DA	0.44223	A	1	0	201
rcv	0.44232	D	401	50	1
snd/DA	0.44232	A	1	0	201
rcv	0.44241	D	451	50	1
snd/DA	0.44241	A	1	0	201
rcv/corr	0.48681	D	200	50	1
...					

```

...
rcv          3336.85176      D      1605201      50      1
snd          3336.85178      A          1      0     1605251
rcv          3336.91281      D      1605301      50      1
snd/DA       3336.91282      A          1      0     1605251
rcv          3336.91284      D      1605351      50      1
snd/DA       3336.91284      A          1      0     1605251
rcv          3336.91286      D      1605401      50      1
snd/DA       3336.91286      A          1      0     1605251
rcv          3336.91288      D      1605401      50      1
snd/DA       3336.91288      A          1      0     1605251
rcv          3336.91289      D      1605451      50      1
snd/DA       3336.91290      A          1      0     1605251
rcv          3336.91291      D      1605551      35      1
snd/DA       3336.91292      A          1      0     1605251
rcv          3336.94534      D      1605251      50      1
snd          3336.94539      A          1      0     1605501
rcv          3337.98002      D      1605501      50      1
snd          3337.98009      A          1      0     1605586
rcv          3337.98014      D      1605501      50      1
snd/DA       3337.98017      A          1      0     1605586
rcv          3337.99564      F      1605586      0      1
snd          3337.99568      A          1      0     1605587
snd          3338.61430      F          1      0     1605587
rcv          3338.63951      A      1605587      0      2

```

```

=====
Amount of data received (bytes)      2069635
Total segments received              41397
Data segments received               41393
Data segments with bit errors        3250
Duplicate data segments received      6031
Duplicate ACKs sent                  27956
=====

```

Appendix E. Logs for Question 5 (c), with no max RTO imposed

Sender_log.txt (snippet)

```

event          time          type      seq      data      ack
snd            36.17522      D        10451     50        1
snd/dup        36.17522      D        10451     50        1
snd/corr       36.17523      D        10551     50        1
snd/corr       36.17524      D        10601     50        1
snd/drop       36.17525      D        10651     50        1
rcv/DA         36.17531      A          1      0     10201
rcv/DA         36.17532      A          1      0     10201
snd/RXT        38.27189      D        10201     50        1
snd/rord       38.27189      D        10501     50        1
rcv            38.28273      A          1      0     10351
rcv/DA         38.28277      A          1      0     10351
snd            38.28283      D        10701     50        1
snd/dup        38.28283      D        10701     50        1
snd            38.28287      D        10751     50        1
snd/dup        38.28287      D        10751     50        1
snd/drop       38.28290      D        10801     50        1
rcv/DA         38.30322      A          1      0     10351
rcv/DA         38.30323      A          1      0     10351
rcv/DA         38.30323      A          1      0     10351
rcv/DA         38.30324      A          1      0     10351
snd/RXT/drop   38.30330      D        10351     50        1
snd/RXT        40.40322      D        10351     50        1
snd/RXT/dup    40.40322      D        10351     50        1
rcv            40.41533      A          1      0     10401
rcv/DA         40.41539      A          1      0     10401

```

snd	40.42321	D	10851	50	1
rcv/DA	40.45121	A	1	0	10401
snd/RXT	42.53581	D	10401	50	1
rcv	42.55937	A	1	0	10551
snd	42.57121	D	10901	50	1
snd	42.57122	D	10951	50	1
snd/dup	42.57122	D	10951	50	1
snd	42.57122	D	11001	50	1
rcv/DA	42.58337	A	1	0	10551
rcv/DA	42.58342	A	1	0	10551
rcv/DA	42.58348	A	1	0	10551
rcv/DA	42.58353	A	1	0	10551
snd/RXT/corr	42.59521	D	10551	50	1
snd/RXT/drop	50.29659	D	10551	50	1
snd/RXT	58.03369	D	10551	50	1
rcv	58.05288	A	1	0	10601
snd	58.05291	D	11051	50	1
rcv/DA	58.07522	A	1	0	10601
snd/RXT	65.79004	D	10601	50	1
rcv	65.79932	A	1	0	10651
snd	65.81121	D	11101	50	1
rcv/DA	65.82332	A	1	0	10651
snd/RXT	81.27363	D	10651	50	1
rcv	81.27382	A	1	0	10801
snd	81.27383	D	11151	50	1
snd/corr	81.27390	D	11201	50	1
snd	81.27391	D	11251	50	1

Receiver_log.txt (snippet)

rcv	36.47907	D	10451	50	1
snd/DA	36.47907	A	1	0	10201
rcv	36.47909	D	10451	50	1
snd/DA	36.47909	A	1	0	10201
rcv/corr	36.47910	D	10551	50	1
rcv/corr	36.47910	D	10601	50	1
rcv	38.57576	D	10201	50	1
snd	38.57581	A	1	0	10351
rcv	38.57583	D	10501	50	1
snd/DA	38.57583	A	1	0	10351
rcv	38.59654	D	10701	50	1
snd/DA	38.59654	A	1	0	10351
rcv	38.59656	D	10701	50	1
snd/DA	38.59656	A	1	0	10351
rcv	38.59657	D	10751	50	1
snd/DA	38.59657	A	1	0	10351
rcv	38.59658	D	10751	50	1
snd/DA	38.59658	A	1	0	10351
rcv	40.71903	D	10351	50	1
snd	40.71908	A	1	0	10401
rcv	40.71910	D	10351	50	1
snd/DA	40.71910	A	1	0	10401
rcv	40.74207	D	10851	50	1
snd/DA	40.74212	A	1	0	10401
rcv	42.86300	D	10401	50	1
snd	42.86301	A	1	0	10551
rcv	42.88703	D	10901	50	1
snd/DA	42.88704	A	1	0	10551
rcv	42.88706	D	10951	50	1
snd/DA	42.88706	A	1	0	10551
rcv	42.88707	D	10951	50	1
snd/DA	42.88707	A	1	0	10551
rcv	42.88708	D	11001	50	1
snd/DA	42.88708	A	1	0	10551
rcv/corr	42.91779	D	10551	50	1
rcv	58.34845	D	10551	50	1

snd	58.34858	A	1	0	10601
rcv	58.36706	D	11051	50	1
snd/DA	58.36710	A	1	0	10601
rcv	66.09393	D	10601	50	1
snd	66.10299	A	1	0	10651
rcv	66.12700	D	11101	50	1
snd/DA	66.12701	A	1	0	10651
rcv	81.57746	D	10651	50	1
snd	81.57750	A	1	0	10801
rcv	81.59900	D	11151	50	1
snd/DA	81.59900	A	1	0	10801
rcv/corr	81.59902	D	11201	50	1
rcv	81.59902	D	11251	50	1