

Assignment 1 report

Tong Chen

5103316

Summary of sender and receiver program implementation:

For this assignment, I finished the standard version. I have finished all the requirements for both sender and receiver. Sender and receiver uses a UDP socket to send data, but maintained a reliable data transfer. Firstly, receiver opens the listening socket, then receiver start the connection establish. After that, receiver reads the data from file, and each time, it extracts the data less than MSS, and packet with segment header, sends to receiver. Sender sends MWS/MSS number of packets to receiver at a time, and waits for the ACK. If the ACK equals to the smallest sequence number plus the data length, the window right shift one packet, and sender sends another packet. If the ACK received is greater than the smallest sequence number plus MWS, the window right shifts $(ACK - Send_Base) / MWS$. If the ACK received is less than the smallest sequence number plus MWS, it retransmits the packet that the ACK indicates. Also, sender maintains a timeout for the oldest sequence number. If it times out, it retransmits the oldest packets in the window. For the receiver, it has a buffer to store the out of order packets, and sends the cumulative ACK. After the file has been sent, sender initiate connection termination, and finally close the socket. I followed the above logic to implement the Sender and Receiver Program. All the features for Sender and Receiver are list below.

1: The three-way shake for the connection establishment has been implemented, with SYN, SYN+ACK, ACK, which can be seen in the sender log file.

2: The four-segment connection termination has been implemented, with (FIN, FIN+ACK, ACK), which can be seen in the

sender log file. I combine the receiver FIN and ACK as one header, so receiver just need to send the FIN and ACK once.

3: Sender has maintained a single-timer for timeout operation. The timer sets timeout for the packet with smallest sequence number in the window, if the sender does not receive the ack within certain time, it will retransmit that particular packet with smallest sequence number.

4: Sender has all the features mentioned in Section 3.5.4. Sender can read data from file, and encapsulates the data in segment with header, each segment includes a sequence number. Sender has timeout. Sender can handle the arrival of an acknowledgement, and compares the ACK value with its SendBase which is the oldest unacknowledged byte. Sender has fast retransmit feature.

5: Receiver has all the features in Section 3.5.4. I do not implement the delayed ACKs.

6: STP is a byte-stream oriented protocol. I have included sequence number and acknowledgement number fields in the STP header for each segment which has the same meaning of TCP sequence number.

7: Sender can deal with different values of MSS, which is a argument in the sender program.

8: Sender has the input argument for MWS, which only counts data, which can deal with dropping packet.

10: I have implemented PLD module as part of my Sender program.

11: Timeout can be supplied to Sender as an input argument.

12: Segment header has two compulsory fields, sequence number and acknowledgement number. Also, header has SYN and FIN flags for connection establishment and teardown. The data portion is less or equal than MSS bytes of data. All the segment for data transferring and acknowledgements are the same, except ACK from receiver does not contain data. I will

explain my header in details latter.

13: Sender has eight(8) arguments followed the requirement.

14: PLD Module is in Sender program; it will not drop segment for connection establishment or teardown.

15: Once all the file has been transmitted, sender start to tear down the connection and finally close the socket. In order to make implementation easier, each packet sent has size MSS, even if the last one does not have data of size MSS, I assume the size is MSS.

16: Sender and receiver both maintained a log file to records the information about each segment that it sends and receives.

17: Receiver can accept two arguments. Receiver ACK immediately after receiving data segment. Receiver has a buffer to buffer the out pf order arrival packets. Receiver creates a new text file called file.txt after receiving the data.

Explanation of STP header:

Sequence	ACK	SYN flag	FIN flag	&#
----------	-----	----------	----------	----

The STP header I used is shown as above. All the header has sequence number and ACK. For connection establishment, I create the header with same format but with a SYN String inside. Also for termination, I keep the same format for header but with a FIN flag inside. For the data transferring state, the header has no SYN or FIN inside, just sequence number and ACK. Finally, I used &# as a separator from header and real data.

Experiment with drop rate and time out:

In order to figure out a suitable value for time out, a set of experiment has been done. (run on test1.txt)

Timeout	Finish time	Packets Dropped	Retransmitted Segments	Duplicate Acknowledgement received
40	82	2	2	10
100	147	2	2	10
200	243	2	2	10
400	436	2	2	10

As the table shows, when time out = 40ms is a suitable value. The time out does not play an important role on the number of packets dropped, retransmitted segments and duplicated acknowledgement received. It only increased the finish time. I will use time out = 40ms for the rest experiment.

Experiment with different drop rate:

(a) Drop rate = 0.1:

Drop rate	Finish time	Packets Dropped	Retransmitted Segments	Duplicate Acknowledgement received
0.1	92	2	2	10
0.3	361	10	10	20

As we can see, when the drop rate increase from 0.1 to 0.3, the number of dropped packet increase from 2 to 10, and also the time for transferring file increased from 50ms to 361ms. The drop rate has an important role on influencing all those parameters. The details will be shown in Appendix.

(b)

Tcurent:

Timeout	Finish time	Packets Dropped	Retransmitted Segments	Duplicate Acknowledgement received
Tcurent	126	4	4	15
4 X Tcurent	374	4	4	15
Tcurent/4	69	4	4	15

Comparing these three experiments, time out does not play a major role on influencing the result. Still, when time out decreased to **Tcurent/4** performed best. When increase the time out value, it seems that no other performance have been influenced except the finish time.

So, **Tcurent/4 will be a good candidate for time out value.**

Appendix:

**(a) Pdrop = 0.1 TEXT1 MSW = 500, MSS = 50, timeout =
40ms seed = 300;**

```
snd  0  S  43 0  0
rcv4  SA 90 0  44
snd  4  A  44 0  91
snd  4  D  44 50 91
snd  5  D  94 50 91
snd  5  D  144  50 91
snd  5  D  194  50 91
snd  5  D  244  50 91
snd  5  D  294  50 91
snd  6  D  344  50 91
snd  6  D  394  50 91
snd  6  D  444  50 91
snd  6  D  494  50 91
rcv9  A  90 0  94
snd  9  D  544  50 91
rcv10 A  90 0  144
snd  10 D  594  50 91
rcv10 A  90 0  194
snd  11 D  644  50 91
rcv11 A  90 0  244
drop  11 D  694  50 91
rcv12 A  90 0  294
snd  12 D  744  50 91
rcv12 A  90 0  344
snd  13 D  794  50 91
rcv13 A  90 0  394
snd  13 D  844  50 91
rcv13 A  90 0  444
snd  13 D  894  50 91
rcv14 A  90 0  494
```

snd 14 D 944 50 91
rcv15 A 90 0 544
snd 16 D 994 50 91
rcv17 A 90 0 594
snd 17 D 1044 50 91
rcv17 A 90 0 644
snd 17 D 1094 50 91
rcv18 A 90 0 694
snd 19 D 1144 50 91
rcv19 A 90 0 694
rcv20 A 90 0 694
rcv21 A 90 0 694
snd 21 D 694 50 91
rcv21 A 90 0 694
rcv22 A 90 0 694
rcv23 A 90 0 694
rcv23 A 90 0 694
rcv24 A 90 0 694
rcv25 A 90 0 694
rcv25 A 90 0 1194
snd 26 D 1194 50 91
snd 26 D 1244 50 91
snd 26 D 1294 50 91
snd 26 D 1344 50 91
snd 26 D 1394 50 91
snd 26 D 1444 50 91
snd 26 D 1494 50 91
drop 26 D 1544 50 91
snd 27 D 1594 50 91
rcv27 A 90 0 1244
rcv28 A 90 0 1294
rcv29 A 90 0 1344
rcv30 A 90 0 1394

rcv30 A 90 0 1444
rcv31 A 90 0 1494
rcv32 A 90 0 1544
rcv33 A 90 0 1544
snd 77 D 1544 50 91
rcv79 A 90 0 1644
snd 79 F 1644 0 91
rcv79 FA 90 0 1644
snd 79 A 1645 0 91

Amount of Data Transferred(in bytes) = 1600

Number of Data Segments Sent(excluding retransmissions) = 32

Number of Packets Dropped = 2

Number of Retransmitted Segments = 2

Number of Duplicate Acknowledgements received = 10

Receiver:

rcv0 S 43 0 0
snd 0 SA 90 0 44
rcv2 A 44 0 91
rcv3 D 44 50 91
snd 3 A 90 0 94
rcv4 D 94 50 91
snd 4 A 90 0 144
rcv5 D 144 50 91
snd 5 A 90 0 194
rcv6 D 194 50 91
snd 6 A 90 0 244
rcv7 D 244 50 91
snd 7 A 90 0 294
rcv8 D 294 50 91
snd 8 A 90 0 344
rcv9 D 344 50 91
snd 9 A 90 0 394

rcv9 D 394 50 91
snd 10 A 90 0 444
rcv11 D 444 50 91
snd 11 A 90 0 494
rcv12 D 494 50 91
snd 12 A 90 0 544
rcv13 D 544 50 91
snd 14 A 90 0 594
rcv14 D 594 50 91
snd 15 A 90 0 644
rcv15 D 644 50 91
snd 16 A 90 0 694
rcv16 D 744 50 91
snd 17 A 90 0 694
rcv17 D 794 50 91
snd 18 A 90 0 694
rcv18 D 844 50 91
snd 18 A 90 0 694
rcv19 D 894 50 91
snd 19 A 90 0 694
rcv20 D 944 50 91
snd 20 A 90 0 694
rcv20 D 994 50 91
snd 21 A 90 0 694
rcv21 D 1044 50 91
snd 21 A 90 0 694
rcv22 D 1094 50 91
snd 22 A 90 0 694
rcv22 D 1144 50 91
snd 22 A 90 0 694
rcv23 D 694 50 91
snd 23 A 90 0 1194
rcv25 D 1194 50 91

```

snd  25 A  90 0 1244
rcv25 D 1244 50 91
snd  26 A  90 0 1294
rcv26 D 1294 50 91
snd  27 A  90 0 1344
rcv27 D 1344 50 91
snd  27 A  90 0 1394
rcv28 D 1394 50 91
snd  28 A  90 0 1444
rcv29 D 1444 50 91
snd  29 A  90 0 1494
rcv30 D 1494 50 91
snd  30 A  90 0 1544
rcv30 D 1594 50 91
snd  31 A  90 0 1544
rcv76 D 1544 50 91
rcv79 F 1644 0 91
snd  79 FA 90 0 1645
rcv79 A 1645 0 91

```

Amount of Data Received(in bytes) = 1600

Number of Data Segments Received = 32

Number of duplicate segments received = 0

(b) Pdrop = 0.3 TEXT1 MSW = 500, MSS = 50, timeout = 40ms seed = 300;

Sender:

```

snd  0 S  43 0 0
rcv4  SA 67 0 44
snd  4 A  44 0 68
snd  5 D  44 50 68
snd  5 D  94 50 68
drop 5 D 144 50 68

```

drop 5 D 194 50 68
snd 5 D 244 50 68
snd 5 D 294 50 68
snd 6 D 344 50 68
snd 6 D 394 50 68
snd 6 D 444 50 68
drop 6 D 494 50 68
rcv8 A 67 0 94
snd 9 D 544 50 68
rcv9 A 67 0 144
snd 9 D 594 50 68
rcv10 A 67 0 144
rcv10 A 67 0 144
rcv11 A 67 0 144
snd 11 D 144 50 68
rcv11 A 67 0 144
rcv12 A 67 0 144
rcv13 A 67 0 144
rcv14 A 67 0 144
rcv14 A 67 0 194
snd 14 D 644 50 68
rcv16 A 67 0 194
snd 57 D 194 50 68
rcv58 A 67 0 494
drop 58 D 694 50 68
snd 59 D 744 50 68
drop 59 D 794 50 68
snd 59 D 844 50 68
snd 59 D 894 50 68
snd 60 D 944 50 68
rcv61 A 67 0 494
rcv61 A 67 0 494
rcv62 A 67 0 494

snd 62 D 494 50 68
rcv63 A 67 0 494
rcv64 A 67 0 694
snd 64 D 994 50 68
drop 64 D 1044 50 68
snd 64 D 1094 50 68
drop 64 D 1144 50 68
rcv65 A 67 0 694
rcv66 A 67 0 694
snd 109 D 694 50 68
rcv111 A 67 0 794
snd 112 D 1194 50 68
drop 112 D 1244 50 68
rcv114 A 67 0 794
snd 156 D 794 50 68
rcv158 A 67 0 1044
snd 158 D 1294 50 68
snd 158 D 1344 50 68
drop 158 D 1394 50 68
snd 158 D 1444 50 68
snd 159 D 1494 50 68
rcv160 A 67 0 1044
rcv163 A 67 0 1044
rcv164 A 67 0 1044
snd 164 D 1044 50 68
rcv165 A 67 0 1044
rcv166 A 67 0 1144
drop 166 D 1544 50 68
snd 166 D 1594 50 68
rcv167 A 67 0 1144
snd 209 D 1144 50 68
rcv211 A 67 0 1244
snd 252 D 1244 50 68

rcv253 A 67 0 1394
snd 294 D 1394 50 68
rcv296 A 67 0 1544
snd 338 D 1544 50 68
rcv339 A 67 0 1644
snd 339 F 1644 0 68
rcv340 FA 67 0 1644
snd 340 A 1645 0 68

Amount of Data Transferred(in bytes) = 1600

Number of Data Segments Sent(excluding retransmissions) = 32

Number of Packets Dropped = 10

Number of Retransmitted Segments = 10

Number of Duplicate Acknowledgements received = 20

Receiver:

rcv0 S 43 0 0
snd 0 SA 67 0 44
rcv1 A 44 0 68
rcv2 D 44 50 68
snd 3 A 67 0 94
rcv3 D 94 50 68
snd 4 A 67 0 144
rcv4 D 244 50 68
snd 5 A 67 0 144
rcv6 D 294 50 68
snd 6 A 67 0 144
rcv7 D 344 50 68
snd 7 A 67 0 144
rcv7 D 394 50 68
snd 8 A 67 0 144
rcv8 D 444 50 68
snd 8 A 67 0 144
rcv9 D 544 50 68

snd 9 A 67 0 144
rcv10 D 594 50 68
snd 10 A 67 0 144
rcv11 D 144 50 68
snd 11 A 67 0 194
rcv12 D 644 50 68
snd 12 A 67 0 194
rcv54 D 194 50 68
snd 55 A 67 0 494
rcv57 D 744 50 68
snd 57 A 67 0 494
rcv58 D 844 50 68
snd 58 A 67 0 494
rcv59 D 894 50 68
snd 59 A 67 0 494
rcv59 D 944 50 68
snd 60 A 67 0 494
rcv60 D 494 50 68
snd 60 A 67 0 694
rcv62 D 994 50 68
snd 62 A 67 0 694
rcv62 D 1094 50 68
snd 62 A 67 0 694
rcv107 D 694 50 68
snd 108 A 67 0 794
rcv110 D 1194 50 68
snd 111 A 67 0 794
rcv154 D 794 50 68
snd 154 A 67 0 1044
rcv156 D 1294 50 68
snd 157 A 67 0 1044
rcv158 D 1344 50 68
snd 159 A 67 0 1044

```

rcv160  D  1444  50 68
snd   160  A  67 0  1044
rcv161  D  1494  50 68
snd   161  A  67 0  1044
rcv162  D  1044  50 68
snd   162  A  67 0  1144
rcv163  D  1594  50 68
snd   163  A  67 0  1144
rcv207  D  1144  50 68
snd   207  A  67 0  1244
rcv250  D  1244  50 68
snd   250  A  67 0  1394
rcv292  D  1394  50 68
snd   293  A  67 0  1544
rcv336  D  1544  50 68
rcv338  F  1644  0  68
snd   339  FA 67 0  1645
rcv339  A  1645  0  68

```

Amount of Data Received(in bytes) = 1600

Number of Data Segments Received = 32

Number of duplicate segments received = 0

(b)

Tcurrent:

Sender:

```

snd   0  S  43 0  0
rcv4  SA 31 0  44
snd   4  A  44 0  32
snd   5  D  44 50 32
snd   5  D  94 50 32
snd   5  D  144 50 32
snd   5  D  194 50 32
snd   6  D  244 50 32

```

snd 6 D 294 50 32
snd 6 D 344 50 32
snd 6 D 394 50 32
snd 6 D 444 50 32
snd 6 D 494 50 32
rcv9 A 31 0 94
snd 9 D 544 50 32
rcv10 A 31 0 144
snd 10 D 594 50 32
rcv10 A 31 0 194
snd 11 D 644 50 32
rcv11 A 31 0 244
drop 12 D 694 50 32
rcv12 A 31 0 294
snd 12 D 744 50 32
rcv13 A 31 0 344
snd 13 D 794 50 32
rcv13 A 31 0 394
snd 14 D 844 50 32
rcv14 A 31 0 444
snd 14 D 894 50 32
rcv14 A 31 0 494
snd 15 D 944 50 32
rcv15 A 31 0 544
snd 15 D 994 50 32
rcv16 A 31 0 594
snd 16 D 1044 50 32
rcv16 A 31 0 644
snd 17 D 1094 50 32
rcv17 A 31 0 694
snd 17 D 1144 50 32
rcv18 A 31 0 694
rcv18 A 31 0 694

rcv19 A 31 0 694
snd 19 D 694 50 32
rcv20 A 31 0 694
rcv20 A 31 0 694
rcv21 A 31 0 694
rcv22 A 31 0 694
rcv22 A 31 0 694
rcv23 A 31 0 694
rcv24 A 31 0 1194
snd 24 D 1194 50 32
snd 25 D 1244 50 32
snd 25 D 1294 50 32
snd 25 D 1344 50 32
snd 25 D 1394 50 32
snd 25 D 1444 50 32
snd 25 D 1494 50 32
drop 25 D 1544 50 32
snd 25 D 1594 50 32
snd 26 D 1644 50 32
rcv26 A 31 0 1244
snd 26 D 1694 50 32
rcv27 A 31 0 1294
drop 27 D 1744 50 32
rcv27 A 31 0 1344
snd 27 D 1794 50 32
rcv28 A 31 0 1394
snd 28 D 1844 50 32
rcv28 A 31 0 1444
drop 28 D 1894 50 32
rcv29 A 31 0 1494
snd 29 D 1944 50 32
rcv29 A 31 0 1544
rcv30 A 31 0 1544

```

rcv30 A 31 0 1544
rcv31 A 31 0 1544
snd 31 D 1544 50 32
rcv31 A 31 0 1544
rcv32 A 31 0 1544
rcv32 A 31 0 1544
rcv33 A 31 0 1744
snd 77 D 1744 50 32
rcv78 A 31 0 1894
snd 120 D 1894 50 32
rcv121 A 31 0 1994
snd 121 F 1994 0 32
rcv121 FA 31 0 1994
snd 121 A 1995 0 32

```

Amount of Data Transferred(in bytes) = 1950

Number of Data Segments Sent(excluding retransmissions) = 39

Number of Packets Dropped = 4

Number of Retransmitted Segments = 4

Number of Duplicate Acknowledgements received = 15

Receiver:

```

rcv0 S 43 0 0
snd 0 SA 31 0 44
rcv1 A 44 0 32
rcv3 D 44 50 32
snd 3 A 31 0 94
rcv4 D 94 50 32
snd 4 A 31 0 144
rcv4 D 144 50 32
snd 5 A 31 0 194
rcv5 D 194 50 32
snd 6 A 31 0 244
rcv6 D 244 50 32

```

snd 6 A 31 0 294
rcv7 D 294 50 32
snd 7 A 31 0 344
rcv8 D 344 50 32
snd 8 A 31 0 394
rcv9 D 394 50 32
snd 9 A 31 0 444
rcv10 D 444 50 32
snd 10 A 31 0 494
rcv11 D 494 50 32
snd 11 A 31 0 544
rcv11 D 544 50 32
snd 12 A 31 0 594
rcv12 D 594 50 32
snd 12 A 31 0 644
rcv13 D 644 50 32
snd 13 A 31 0 694
rcv14 D 744 50 32
snd 14 A 31 0 694
rcv15 D 794 50 32
snd 15 A 31 0 694
rcv15 D 844 50 32
snd 15 A 31 0 694
rcv16 D 894 50 32
snd 16 A 31 0 694
rcv17 D 944 50 32
snd 17 A 31 0 694
rcv18 D 994 50 32
snd 18 A 31 0 694
rcv18 D 1044 50 32
snd 18 A 31 0 694
rcv19 D 1094 50 32
snd 19 A 31 0 694

rcv20 D 1144 50 32
snd 20 A 31 0 694
rcv21 D 694 50 32
snd 21 A 31 0 1194
rcv22 D 1194 50 32
snd 23 A 31 0 1244
rcv23 D 1244 50 32
snd 23 A 31 0 1294
rcv24 D 1294 50 32
snd 24 A 31 0 1344
rcv24 D 1344 50 32
snd 24 A 31 0 1394
rcv25 D 1394 50 32
snd 25 A 31 0 1444
rcv25 D 1444 50 32
snd 26 A 31 0 1494
rcv26 D 1494 50 32
snd 26 A 31 0 1544
rcv27 D 1594 50 32
snd 27 A 31 0 1544
rcv27 D 1644 50 32
snd 27 A 31 0 1544
rcv27 D 1694 50 32
snd 28 A 31 0 1544
rcv28 D 1794 50 32
snd 28 A 31 0 1544
rcv28 D 1844 50 32
snd 28 A 31 0 1544
rcv29 D 1944 50 32
snd 29 A 31 0 1544
rcv29 D 1544 50 32
snd 29 A 31 0 1744
rcv74 D 1744 50 32

snd 75 A 31 0 1894
rcv117 D 1894 50 32
rcv120 F 1994 0 32
snd 120 FA 31 0 1995
rcv120 A 1995 0 32

Amount of Data Received(in bytes) = 1950

Number of Data Segments Received = 39

Number of duplicate segments received = 0

4 X Tcurrent:

Sender:

snd 0 S 43 0 0
rcv4 SA 12 0 44
snd 4 A 44 0 13
snd 6 D 44 50 13
snd 6 D 94 50 13
snd 6 D 144 50 13
snd 6 D 194 50 13
snd 6 D 244 50 13
snd 6 D 294 50 13
snd 6 D 344 50 13
snd 7 D 394 50 13
snd 7 D 444 50 13
snd 7 D 494 50 13
rcv9 A 12 0 94
snd 10 D 544 50 13
rcv11 A 12 0 144
snd 11 D 594 50 13
rcv11 A 12 0 194
snd 11 D 644 50 13
rcv12 A 12 0 244
drop 12 D 694 50 13

rcv12 A 12 0 294
snd 13 D 744 50 13
rcv13 A 12 0 344
snd 14 D 794 50 13
rcv14 A 12 0 394
snd 14 D 844 50 13
rcv14 A 12 0 444
snd 15 D 894 50 13
rcv16 A 12 0 494
snd 16 D 944 50 13
rcv17 A 12 0 544
snd 17 D 994 50 13
rcv19 A 12 0 594
snd 19 D 1044 50 13
rcv20 A 12 0 644
snd 20 D 1094 50 13
rcv20 A 12 0 694
snd 20 D 1144 50 13
rcv21 A 12 0 694
rcv22 A 12 0 694
rcv22 A 12 0 694
snd 23 D 694 50 13
rcv23 A 12 0 694
rcv24 A 12 0 694
rcv24 A 12 0 694
rcv25 A 12 0 694
rcv26 A 12 0 694
rcv26 A 12 0 694
rcv27 A 12 0 1194
snd 27 D 1194 50 13
snd 27 D 1244 50 13
snd 27 D 1294 50 13
snd 27 D 1344 50 13

snd 28 D 1394 50 13
snd 28 D 1444 50 13
snd 29 D 1494 50 13
drop 29 D 1544 50 13
snd 29 D 1594 50 13
snd 29 D 1644 50 13
rcv30 A 12 0 1244
snd 30 D 1694 50 13
rcv30 A 12 0 1294
drop 30 D 1744 50 13
rcv31 A 12 0 1344
snd 31 D 1794 50 13
rcv31 A 12 0 1394
snd 31 D 1844 50 13
rcv32 A 12 0 1444
drop 32 D 1894 50 13
rcv32 A 12 0 1494
snd 33 D 1944 50 13
rcv33 A 12 0 1544
rcv34 A 12 0 1544
rcv35 A 12 0 1544
rcv38 A 12 0 1544
snd 38 D 1544 50 13
rcv39 A 12 0 1544
rcv42 A 12 0 1544
rcv43 A 12 0 1544
rcv44 A 12 0 1744
snd 206 D 1744 50 13
rcv207 A 12 0 1894
snd 370 D 1894 50 13
rcv372 A 12 0 1994
snd 372 F 1994 0 13
rcv372 FA 12 0 1994

snd 372 A 1995 0 13

Amount of Data Transferred(in bytes) = 1950

Number of Data Segments Sent(excluding retransmissions) = 39

Number of Packets Dropped = 4

Number of Retransmitted Segments = 4

Number of Duplicate Acknowledgements received = 15

Receiver:

rcv0 S 43 0 0

snd 0 SA 12 0 44

rcv2 A 44 0 13

rcv3 D 44 50 13

snd 4 A 12 0 94

rcv4 D 94 50 13

snd 5 A 12 0 144

rcv5 D 144 50 13

snd 6 A 12 0 194

rcv6 D 194 50 13

snd 6 A 12 0 244

rcv7 D 244 50 13

snd 7 A 12 0 294

rcv8 D 294 50 13

snd 8 A 12 0 344

rcv9 D 344 50 13

snd 9 A 12 0 394

rcv10 D 394 50 13

snd 10 A 12 0 444

rcv12 D 444 50 13

snd 12 A 12 0 494

rcv13 D 494 50 13

snd 14 A 12 0 544

rcv15 D 544 50 13

snd 15 A 12 0 594

rcv16 D 594 50 13
snd 16 A 12 0 644
rcv17 D 644 50 13
snd 17 A 12 0 694
rcv17 D 744 50 13
snd 18 A 12 0 694
rcv18 D 794 50 13
snd 19 A 12 0 694
rcv19 D 844 50 13
snd 19 A 12 0 694
rcv20 D 894 50 13
snd 20 A 12 0 694
rcv20 D 944 50 13
snd 21 A 12 0 694
rcv21 D 994 50 13
snd 21 A 12 0 694
rcv22 D 1044 50 13
snd 22 A 12 0 694
rcv22 D 1094 50 13
snd 22 A 12 0 694
rcv23 D 1144 50 13
snd 23 A 12 0 694
rcv24 D 694 50 13
snd 24 A 12 0 1194
rcv25 D 1194 50 13
snd 25 A 12 0 1244
rcv26 D 1244 50 13
snd 26 A 12 0 1294
rcv27 D 1294 50 13
snd 27 A 12 0 1344
rcv28 D 1344 50 13
snd 28 A 12 0 1394
rcv28 D 1394 50 13

```

snd  28 A  12 0  1444
rcv29 D 1444 50 13
snd  29 A  12 0  1494
rcv29 D 1494 50 13
snd  30 A  12 0  1544
rcv30 D 1594 50 13
snd  30 A  12 0  1544
rcv31 D 1644 50 13
snd  31 A  12 0  1544
rcv35 D 1694 50 13
snd  35 A  12 0  1544
rcv36 D 1794 50 13
snd  36 A  12 0  1544
rcv38 D 1844 50 13
snd  38 A  12 0  1544
rcv39 D 1944 50 13
snd  39 A  12 0  1544
rcv40 D 1544 50 13
snd  40 A  12 0  1744
rcv204 D 1744 50 13
snd  204 A  12 0  1894
rcv368 D 1894 50 13
rcv371 F 1994 0 13
snd  372 FA 12 0 1995
rcv372 A 1995 0 13

```

Amount of Data Received(in bytes) = 1950

Number of Data Segments Received = 39

Number of duplicate segments received = 0

Tcurrent/4:

Sender:

```

snd  0 S  43 0  0

```

rcv4 SA 69 0 44
snd 4 A 44 0 70
snd 5 D 44 50 70
snd 5 D 94 50 70
snd 5 D 144 50 70
snd 5 D 194 50 70
snd 6 D 244 50 70
snd 6 D 294 50 70
snd 6 D 344 50 70
snd 6 D 394 50 70
snd 7 D 444 50 70
snd 7 D 494 50 70
rcv10 A 69 0 94
snd 10 D 544 50 70
rcv11 A 69 0 144
snd 11 D 594 50 70
rcv11 A 69 0 194
snd 12 D 644 50 70
rcv13 A 69 0 244
drop 13 D 694 50 70
rcv13 A 69 0 294
snd 13 D 744 50 70
rcv14 A 69 0 344
snd 14 D 794 50 70
rcv14 A 69 0 394
snd 14 D 844 50 70
rcv15 A 69 0 444
snd 16 D 894 50 70
rcv17 A 69 0 494
snd 17 D 944 50 70
rcv19 A 69 0 544
snd 19 D 994 50 70
rcv20 A 69 0 594

snd 21 D 1044 50 70
rcv21 A 69 0 644
snd 22 D 1094 50 70
rcv22 A 69 0 694
snd 22 D 1144 50 70
rcv23 A 69 0 694
rcv24 A 69 0 694
rcv25 A 69 0 694
snd 25 D 694 50 70
rcv26 A 69 0 694
rcv27 A 69 0 694
rcv28 A 69 0 694
rcv28 A 69 0 694
rcv29 A 69 0 694
rcv30 A 69 0 694
rcv31 A 69 0 1194
snd 32 D 1194 50 70
snd 32 D 1244 50 70
snd 32 D 1294 50 70
snd 32 D 1344 50 70
snd 32 D 1394 50 70
snd 32 D 1444 50 70
snd 32 D 1494 50 70
drop 32 D 1544 50 70
snd 32 D 1594 50 70
snd 32 D 1644 50 70
rcv34 A 69 0 1244
snd 34 D 1694 50 70
rcv41 A 69 0 1294
drop 41 D 1744 50 70
rcv44 A 69 0 1344
snd 44 D 1794 50 70
rcv45 A 69 0 1394

```

snd  45 D 1844 50 70
rcv46 A 69 0 1444
drop 46 D 1894 50 70
rcv47 A 69 0 1494
snd  47 D 1944 50 70
rcv47 A 69 0 1544
rcv48 A 69 0 1544
rcv49 A 69 0 1544
rcv50 A 69 0 1544
snd  50 D 1544 50 70
rcv50 A 69 0 1544
rcv51 A 69 0 1544
rcv51 A 69 0 1544
rcv52 A 69 0 1744
snd  65 D 1744 50 70
rcv65 A 69 0 1894
snd  76 D 1894 50 70
rcv80 A 69 0 1994
snd  80 F 1994 0 70
rcv80 FA 69 0 1994
snd  80 A 1995 0 70

```

Amount of Data Transferred(in bytes) = 1950

Number of Data Segments Sent(excluding retransmissions) = 39

Number of Packets Dropped = 4

Number of Retransmitted Segments = 4

Number of Duplicate Acknowledgements received = 15

Receiver:

```

rcv0  S 43 0 0
snd  1  SA 69 0 44
rcv2  A 44 0 70
rcv4  D 44 50 70
snd  4  A 69 0 94

```

rcv5 D 94 50 70
snd 5 A 69 0 144
rcv6 D 144 50 70
snd 6 A 69 0 194
rcv7 D 194 50 70
snd 8 A 69 0 244
rcv9 D 244 50 70
snd 9 A 69 0 294
rcv10 D 294 50 70
snd 10 A 69 0 344
rcv11 D 344 50 70
snd 11 A 69 0 394
rcv12 D 394 50 70
snd 13 A 69 0 444
rcv14 D 444 50 70
snd 14 A 69 0 494
rcv16 D 494 50 70
snd 16 A 69 0 544
rcv18 D 544 50 70
snd 18 A 69 0 594
rcv19 D 594 50 70
snd 19 A 69 0 644
rcv20 D 644 50 70
snd 20 A 69 0 694
rcv21 D 744 50 70
snd 21 A 69 0 694
rcv22 D 794 50 70
snd 22 A 69 0 694
rcv23 D 844 50 70
snd 23 A 69 0 694
rcv23 D 894 50 70
snd 23 A 69 0 694
rcv24 D 944 50 70

snd 24 A 69 0 694
rcv25 D 994 50 70
snd 25 A 69 0 694
rcv26 D 1044 50 70
snd 26 A 69 0 694
rcv27 D 1094 50 70
snd 27 A 69 0 694
rcv28 D 1144 50 70
snd 28 A 69 0 694
rcv29 D 694 50 70
snd 30 A 69 0 1194
rcv31 D 1194 50 70
snd 31 A 69 0 1244
rcv35 D 1244 50 70
snd 36 A 69 0 1294
rcv42 D 1294 50 70
snd 42 A 69 0 1344
rcv42 D 1344 50 70
snd 42 A 69 0 1394
rcv43 D 1394 50 70
snd 43 A 69 0 1444
rcv44 D 1444 50 70
snd 44 A 69 0 1494
rcv45 D 1494 50 70
snd 45 A 69 0 1544
rcv46 D 1594 50 70
snd 46 A 69 0 1544
rcv46 D 1644 50 70
snd 46 A 69 0 1544
rcv47 D 1694 50 70
snd 47 A 69 0 1544
rcv48 D 1794 50 70
snd 48 A 69 0 1544

rcv48 D 1844 50 70
snd 48 A 69 0 1544
rcv49 D 1944 50 70
snd 49 A 69 0 1544
rcv50 D 1544 50 70
snd 50 A 69 0 1744
rcv63 D 1744 50 70
snd 63 A 69 0 1894
rcv75 D 1894 50 70
rcv80 F 1994 0 70
snd 80 FA 69 0 1995
rcv80 A 1995 0 70

Amount of Data Received(in bytes) = 1950

Number of Data Segments Received = 39

Number of duplicate segments received = 0