Report. Z5172023 Jialun Li

1. A brief discussion of how you have implemented the STP protocol. Provide a list of features that you have successfully implemented. In case you have not been able to get certain features of STP working, you should also mention that in your report.

(1) I implement three shake hands based on a send-waiting way. For example, sender start sending SYN, then if and only if the server send SYN AKC and client receive it, the sender would send ACK for connection. Otherwise it would do nothing. While when it comes to transfer the data, I do 4 step in an infinite loop such that 1. Check for timeout 2. Check and collect for received ack from server. 3. Slide window according to the last received ack and enqueue new data to be sent. 4. Sent all the data in window which are never sent.If there is nothing to do in any of these step just simply skip and get into next loop.

(2) I have done all the features it needed [three shake hands, reliable transfer data, four segment tear down, timeout (including calculate and update interval after get each non-retransferred ack), fast retransmition (when sender receive 3 duplicated ACK), slide window protocol (dequeue and enqueue), check_sum for check bit error, cumulative acknowledgement (when slide the window), a well_designed header for all segment, MSS and MWS, all error segment produce in PLD module]

2. A detailed diagram of your STP header and a quick explanation of all fields (similar to the diagrams that we have used in the lectures to understand TCP/UDP headers).

Sequence number	Ack number	Check sum	ACK	SYN	FIN	Data size	File size

The header consist of 8 numbers combined together by a string ‘$’ and encode(‘utf-8’). Finally add with payload.

For example a whole segment like b’301$1$0$0$0$0$150$3028$datadatadatadata’

Sequence number	Ack number	Check sum	ACK	SYN	FIN	Data size	File size
301	1	0	0	0	0	150	3028

Its data datagram is b’datadatadatadata’

The size of my designed header is not fixed as its using utf-8 encode. It means every string counts a bytes.

So, basically the size of header depends on the total length of the strings(number) in header

For example ’1$1$0$0$0$0$150$3028$’ is 20 bytes.

3. Discuss any design trade-offs considered and made. Describe possible improvements and extensions to your program and indicate how you could realise them.

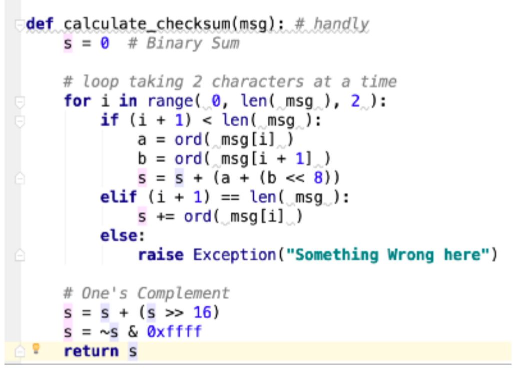
(1) For implementing timer, I check every sent segment for timeout once in an infinite loop. It’s easy to code but it may cause the cpu waste.

(2) One possible improvement for my program is to accomplish it by using multiple threads such that one thread for sending data the other thread for receiving ack from sender. While I used single thread.

To implement multiple threads, I have to import python threading module, and when different threads read or write the same data , there has to be a lock and release operation.

4. Indicate any segments of code that you have borrowed from the Web or other books.

the code for computing the check_sum.



```python
def calculate_checksum(msg): # handly
    s = 0  # Binary Sum

    # loop taking 2 characters at a time
    for i in range( 0, len( msg ), 2 ):
        if (i + 1) < len( msg ):
            a = ord( msg[i] )
            b = ord( msg[i + 1] )
            s = s + (a + (b << 8))
        elif (i + 1) == len( msg ):
            s += ord( msg[i] )
        else:
            raise Exception("Something Wrong here")

    # One's Complement
    s = s + (s >> 16)
    s = ~s & 0xffff
    return s
```

5. Answer the following questions: (include any output as an appendix to the main report.pdf, appendix is not included in the 5-page limit)

(a) Run your protocol using pDrop = 0.1, MWS = 500 bytes, MSS = 100 bytes, seed = 100, gamma = 4, and pDuplicate, pCorrupt, pOrder, MaxOrder, pDelay, MaxDelay all set to 0. Transfer the file **test0.pdf** (available on the assignment webpage). The file should be received correctly at the Receiver. Show the sequence of STP packets that are observed at the Receiver. It is sufficient to just indicate the sequence numbers of the STP packets that have arrived. Run an additional experiment with pdrop = 0.3, transferring the same file (**test0.pdf**). In your report, discuss the resulting packet sequences of both experiments indicating where dropping occurred. Also, in the appendix section show the packet sequences for both the experiments.

(1)For experiments 1, the sequence number of drop packages are 201, 2801, 2901

(2)For experiments 2, the sequence number of drop packages are 1 401 501 801 1601 1801 2101 2401 2501 2601 2701

(b) The timeout for STP is given by:

```
TimeoutInterval = EstimatedRTT + gamma * DevRTT
```

where `gamma` will be supplied to the program as an input argument, see Section 4.5.

Set pdrop = 0.5, MWS = 500 bytes, MSS = 50 bytes, seed = 300, pdelay = 0.2, MaxDelay = 1000 and pDuplicate, pCorrupt, pOrder, MaxOrder all set to 0. Run three experiments with the following different gamma values:

i. gamma = 2

ii. gamma = 4

iii. gamma = 9

and transfer the file **test1.pdf** using STP. Show a table that indicates how many STP packets were transmitted in total and how long the overall transfer took. Discuss the results.

1.

	gamma = 2	gamma = 4	gamma = 9
Total number transferred	283799	43606	776.63
Time	11min	11min	13min

When gamma is increasing the interval to cause a timeout increases as well. It means system will waiting more time for the packages dropped and delayed which takes longer time.

(c) Use the following values and run STP to transfer **test2.pdf**.

`MWS=500bytes MSS=50 gamma=4 pDrop=0.1 pDuplicate=0.1 pCorrupt=0.1`

`pOrder=0.1 maxOrder=4 pDelay=0 maxDelay=0 seed=300`

Has the file been successfully transferred? How long the overall transfer took? For this experiment, which of the factor (out of `pDrop`, `pDuplicate`, `pCorrupt` and `pOrder`) is the most critical contributing most in the overall transfer time? How have you determined this? Provide the sender_log.txt and receiver_log.txt files in appendix.

Yes, the file been successfully transferred.

It took around 150s

`Drop.` Because it occurs the most and every time it occurs there is a interval of timeout to wait.

Appendix for question (a)

(a)

Experiment 1. The sequence number received observed from server:

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

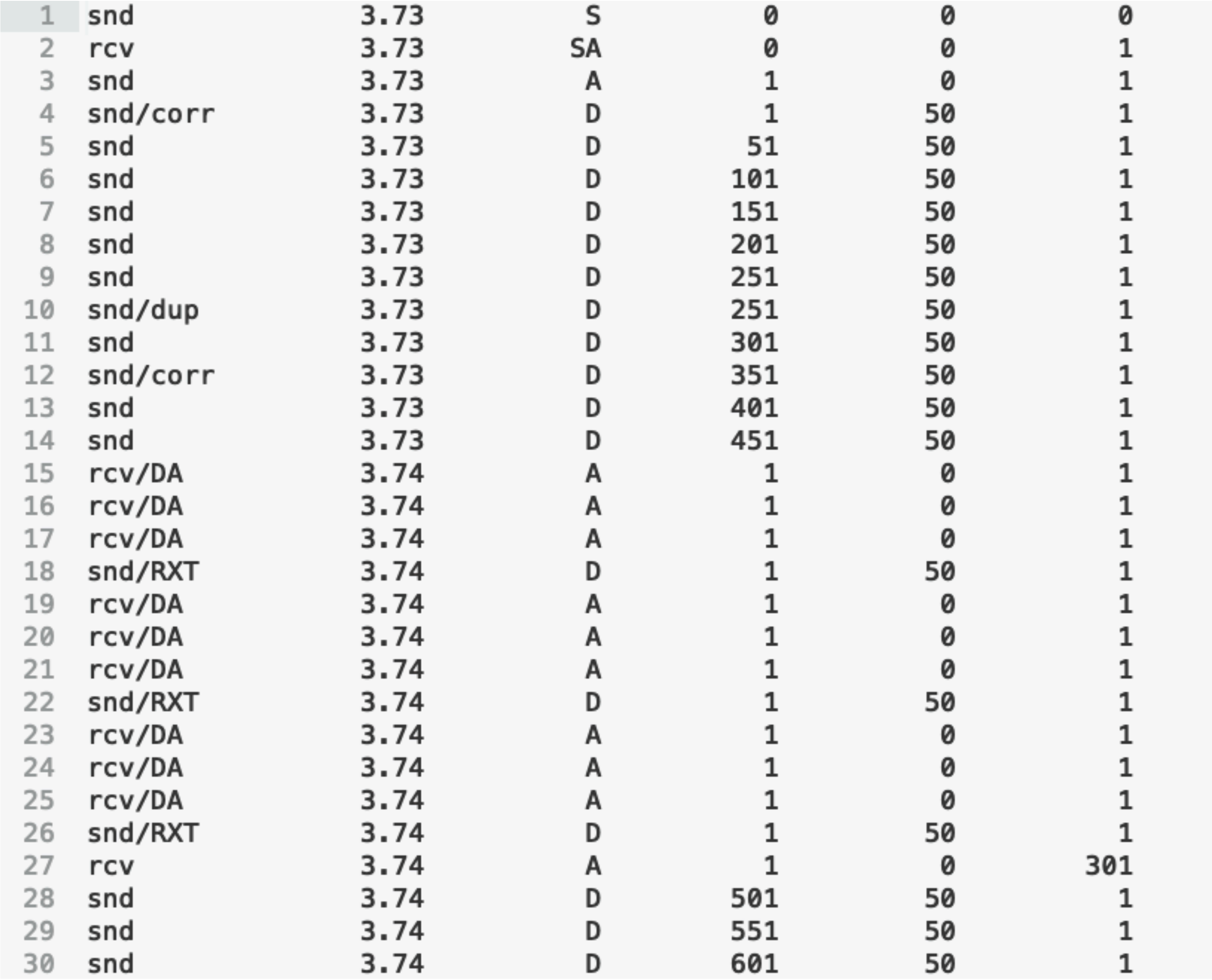
Experiment 2. The sequence number received observed from server:

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

log for question (c)

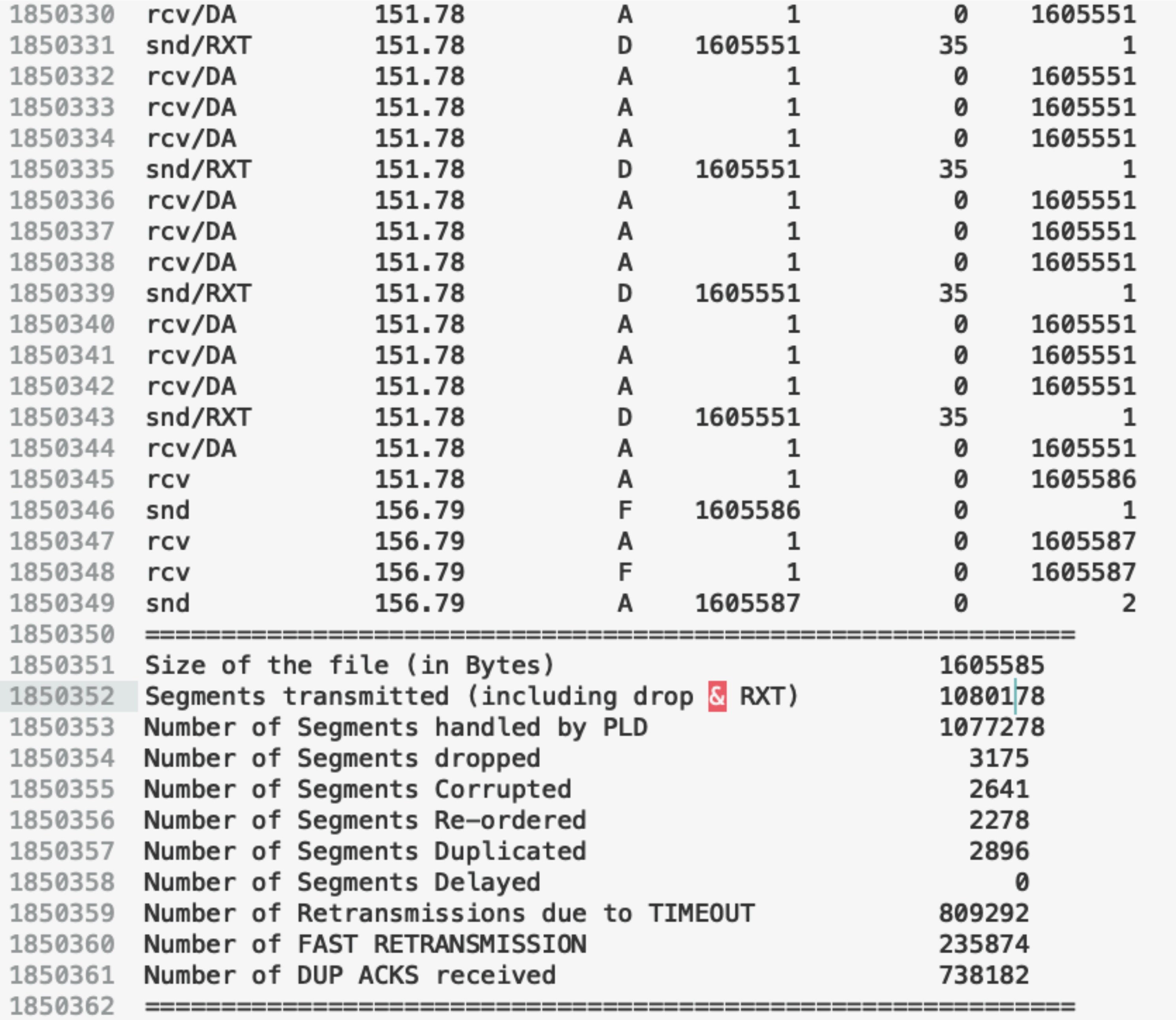
sender

First 20



```
snd                3.73          S         0         0         0
rcv                3.73         SA         0         0         1
snd                3.73          A         1         0         1
snd/corr           3.73          D         1        50         1
snd                3.73          D        51        50         1
snd                3.73          D       101        50         1
snd                3.73          D       151        50         1
snd                3.73          D       201        50         1
snd                3.73          D       251        50         1
snd/dup            3.73          D       251        50         1
snd                3.73          D       301        50         1
snd/corr           3.73          D       351        50         1
snd                3.73          D       401        50         1
snd                3.73          D       451        50         1
rcv/DA             3.74          A         1         0         1
rcv/DA             3.74          A         1         0         1
rcv/DA             3.74          A         1         0         1
snd/RXT            3.74          D         1        50         1
rcv/DA             3.74          A         1         0         1
rcv/DA             3.74          A         1         0         1
rcv/DA             3.74          A         1         0         1
snd/RXT            3.74          D         1        50         1
rcv/DA             3.74          A         1         0         1
rcv/DA             3.74          A         1         0         1
rcv/DA             3.74          A         1         0         1
snd/RXT            3.74          D         1        50         1
rcv                3.74          A         1         0       301
snd                3.74          D       501        50         1
snd                3.74          D       551        50         1
snd                3.74          D       601        50         1
```

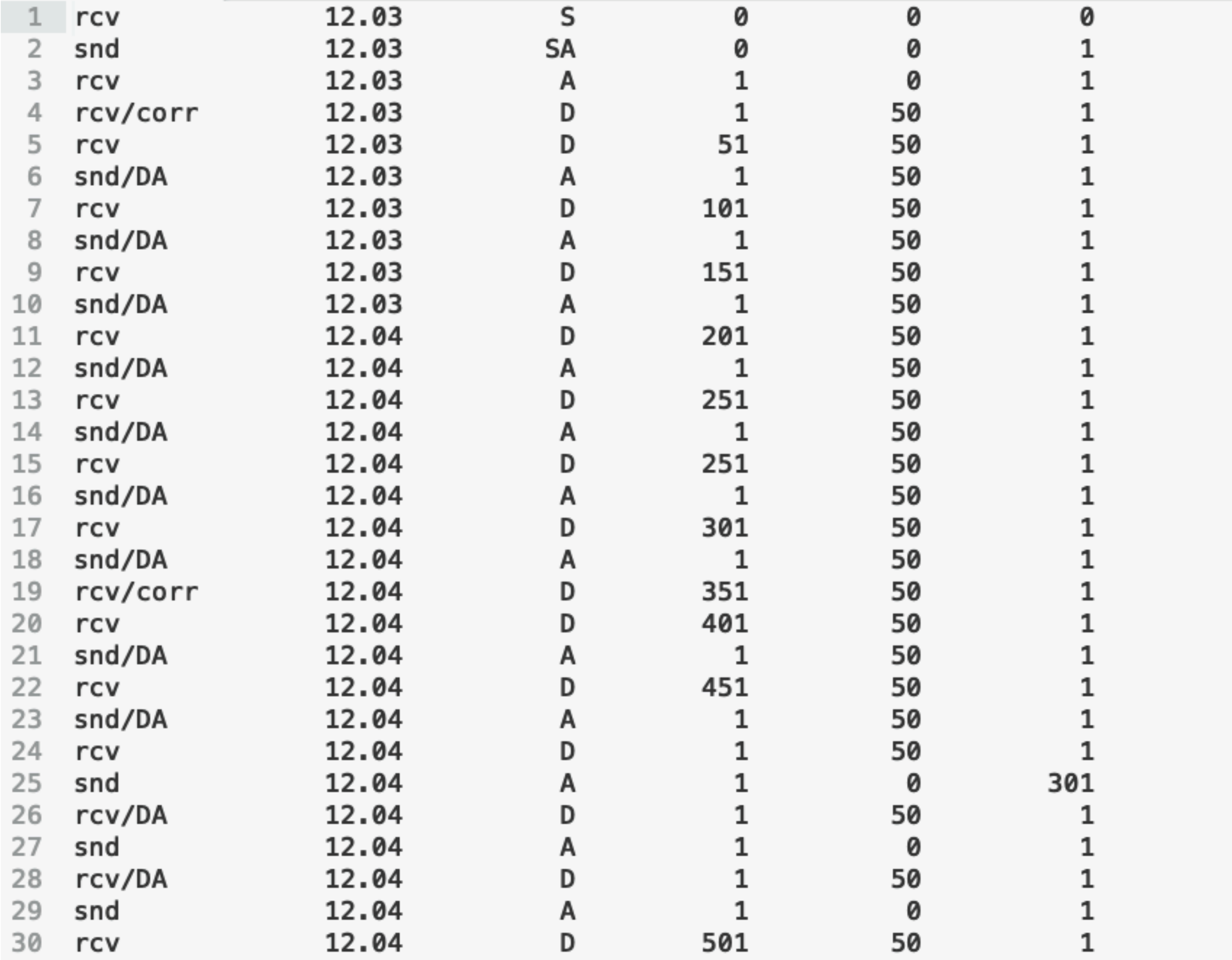
Last 20 and summary



```
rcv/DA         151.78          A          1          0    1605551
snd/RXT        151.78          D    1605551         35          1
rcv/DA         151.78          A          1          0    1605551
rcv/DA         151.78          A          1          0    1605551
rcv/DA         151.78          A          1          0    1605551
snd/RXT        151.78          D    1605551         35          1
rcv/DA         151.78          A          1          0    1605551
rcv/DA         151.78          A          1          0    1605551
rcv/DA         151.78          A          1          0    1605551
snd/RXT        151.78          D    1605551         35          1
rcv/DA         151.78          A          1          0    1605551
rcv/DA         151.78          A          1          0    1605551
rcv/DA         151.78          A          1          0    1605551
snd/RXT        151.78          D    1605551         35          1
rcv/DA         151.78          A          1          0    1605551
rcv            151.78          A          1          0    1605586
snd            156.79          F    1605586          0          1
rcv            156.79          A          1          0    1605587
rcv            156.79          F          1          0    1605587
snd            156.79          A    1605587          0          2
============================================================
Size of the file (in Bytes)                        1605585
Segments transmitted (including drop & RXT)        1080178
Number of Segments handled by PLD                  1077278
Number of Segments dropped                           3175
Number of Segments Corrupted                         2641
Number of Segments Re-ordered                        2278
Number of Segments Duplicated                        2896
Number of Segments Delayed                              0
Number of Retransmissions due to TIMEOUT           809292
Number of FAST RETRANSMISSION                      235874
Number of DUP ACKS received                        738182
============================================================
```

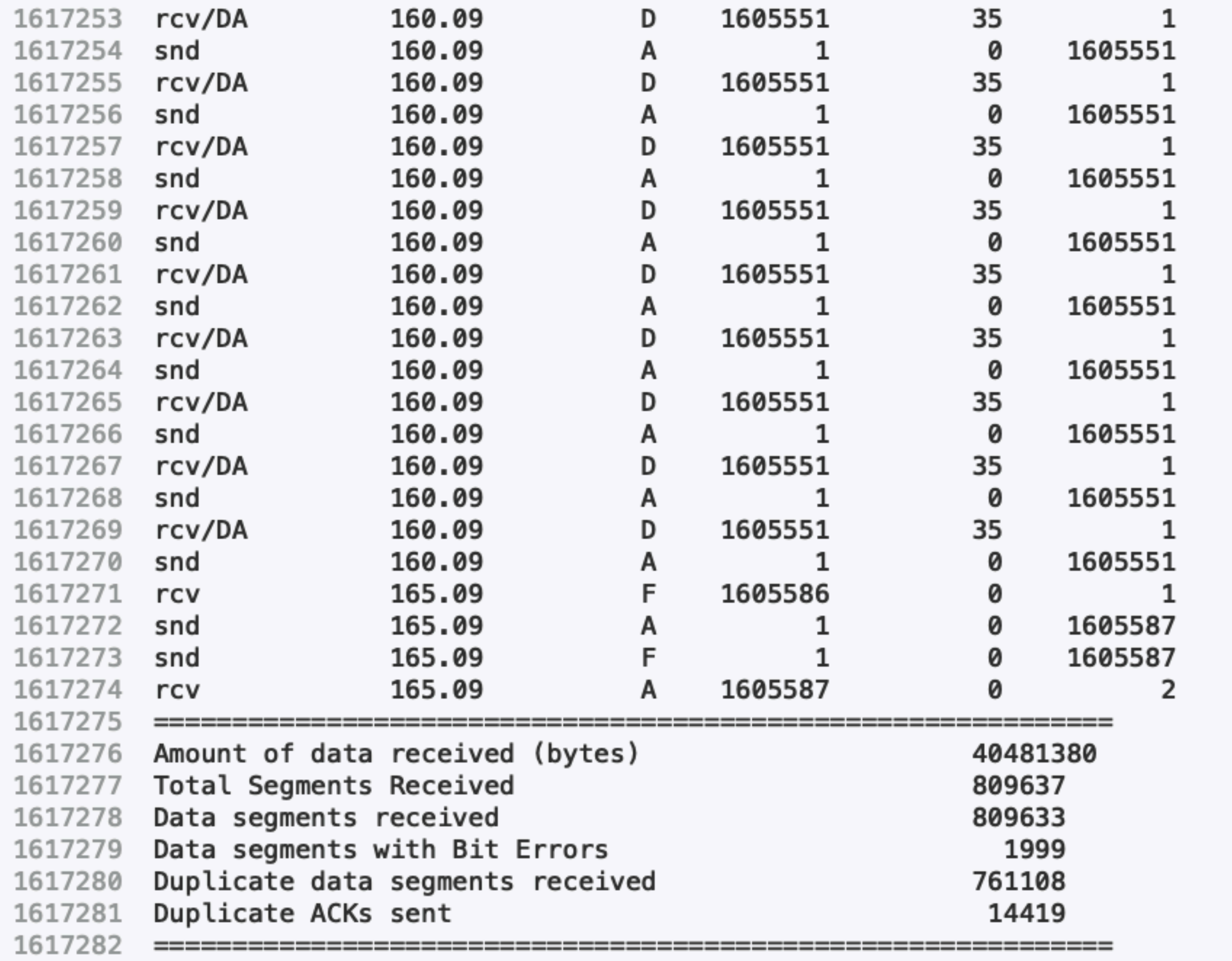
Server

First 20



```
rcv           12.03      S       0      0      0
snd           12.03     SA       0      0      1
rcv           12.03      A       1      0      1
rcv/corr      12.03      D       1     50      1
rcv           12.03      D      51     50      1
snd/DA        12.03      A       1     50      1
rcv           12.03      D     101     50      1
snd/DA        12.03      A       1     50      1
rcv           12.03      D     151     50      1
snd/DA        12.03      A       1     50      1
rcv           12.04      D     201     50      1
snd/DA        12.04      A       1     50      1
rcv           12.04      D     251     50      1
snd/DA        12.04      A       1     50      1
rcv           12.04      D     251     50      1
snd/DA        12.04      A       1     50      1
rcv           12.04      D     301     50      1
snd/DA        12.04      A       1     50      1
rcv/corr      12.04      D     351     50      1
rcv           12.04      D     401     50      1
snd/DA        12.04      A       1     50      1
rcv           12.04      D     451     50      1
snd/DA        12.04      A       1     50      1
rcv           12.04      D       1     50      1
snd           12.04      A       1      0    301
rcv/DA        12.04      D       1     50      1
snd           12.04      A       1      0      1
rcv/DA        12.04      D       1     50      1
snd           12.04      A       1      0      1
rcv           12.04      D     501     50      1
```

Last 20 and summary



```
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv/DA         160.09          D    1605551         35          1
snd            160.09          A          1          0    1605551
rcv            165.09          F    1605586          0          1
snd            165.09          A          1          0    1605587
snd            165.09          F          1          0    1605587
rcv            165.09          A    1605587          0          2
=============================================================
Amount of data received (bytes)                     40481380
Total Segments Received                             809637
Data segments received                              809633
Data segments with Bit Errors                         1999
Duplicate data segments received                    761108
Duplicate ACKs sent                                  14419
=============================================================
```