Eusebius Ballentine

CS 372

2-24-2025

Assignment 2 - Introduction

The bulk of the logic was implemented in the RDTLayer class's processReceiveAndSendRespond() and processSend() functions. In each function, it was essential to differentiate between each of the RDTLayers, client and server, and then to implement the necessary logic within each of their respective code blocks.

Timeouts were resolved by utilizing the RDTLayer class data member self.unAckedIteration in the server instance to track the iterations in the event that a client segment's sequence number did not equal the ack segment sent to the client or in the event of a checksum error. Once the event condition is met, the server appends the ack number from it's self.currentAcknum data member to it's self.serverUnAckedSegList, which tracks unacked client segments. In the function code, there is a condition which checks if incoming segment sequence numbers are in self.serverUnAckedSegList, and if it is, that element is removed from the list.

Packet dropping is handled by the same conditional statement. If a packet is dropped, the starting sequence number of that sequence of bytes in the data would be added to the server's self.serverUnAckedSegList, and if 3 iterations with out of order sequence bytes, the server would ack with a sequence number from it's self.serverUnAckedSegList containing the lost packets.

Retransmission was implemented by assigning the first element in it's self.serverUnAckedSegList to the acknum member of it's ack segment. This implementation is more of a selective retransmit version of TCP because self.serverUnAckedSegList contains an unordered list of sequence numbers of packets. The list becomes unordered as lost or delayed packets are removed from the list and the server pulls unacked sequence numbers from the front of the list, which should be one of the oldest unacked segments because sequence numbers are appended to the end of the list.

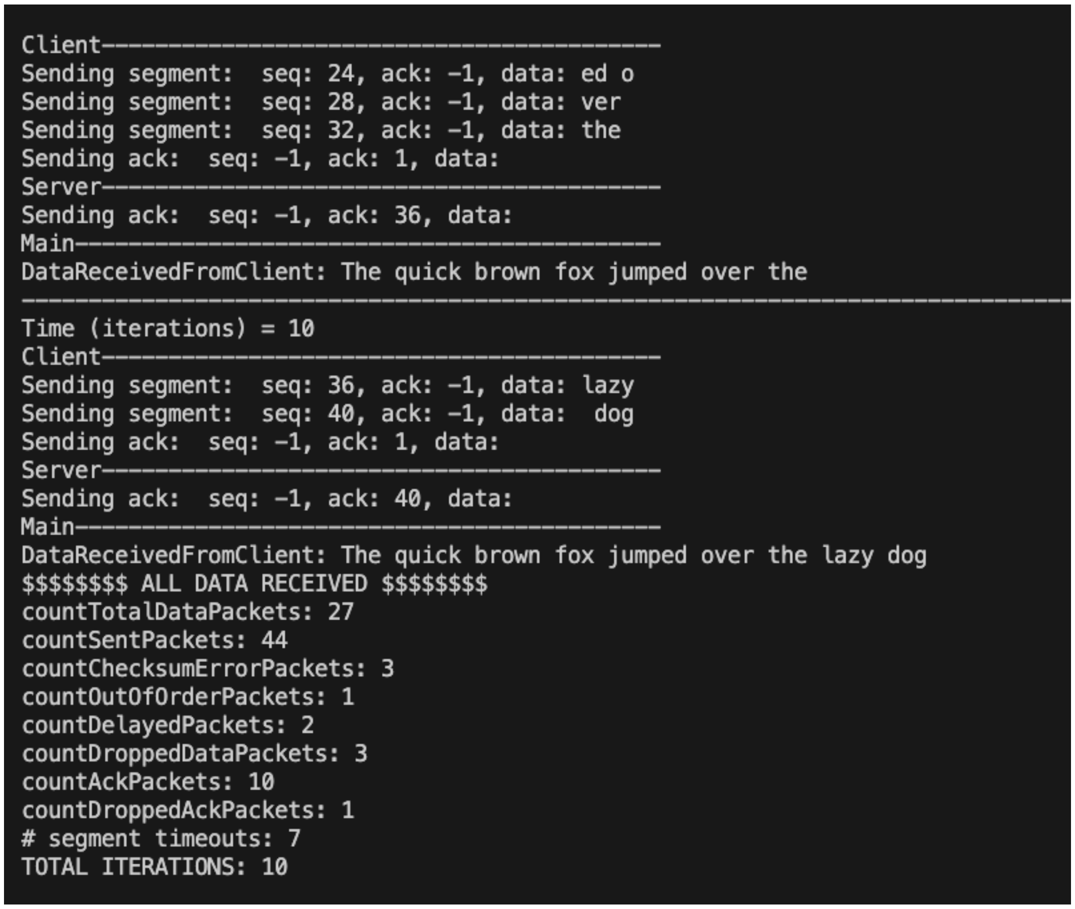
Note: I did change the rdt_main.py file to name the RDTLayer instances, which I used to differentiate between client and server in the processSend() and processReceiveAndSendRespond() functions. It may not have been a preferred method for identification, but I created constant variables to make the code a bit more modular instead of using strings while instantiating the RDTLayer objects.

Summary:

This program ended up not being very efficient and I discovered that self.serverUnAckedSegList was not exhibiting the intended behavior. I can see that the conditional statement that was supposed to catch packets with out of order sequence numbers or checksum errors was not designed properly, and as a result, self.serverUnAckedSegList was not efficiently adding and removing sequence numbers, thus decreasing efficiency. Retransmission happened effectively, but in a very inefficient way as evidenced by the clients self.countSegmentTimeout count, as well as countSentPackets and Total iterations. However, the program was able to transmit the data accurately with all unreliable flags set in the UnreliableChannel instances. Ultimately, I felt like the project got messy and a bit out of hand, and then I ran out of time. In the future, I think that I need to start the project earlier and more planning should be done prior to coding, which is almost always the case.

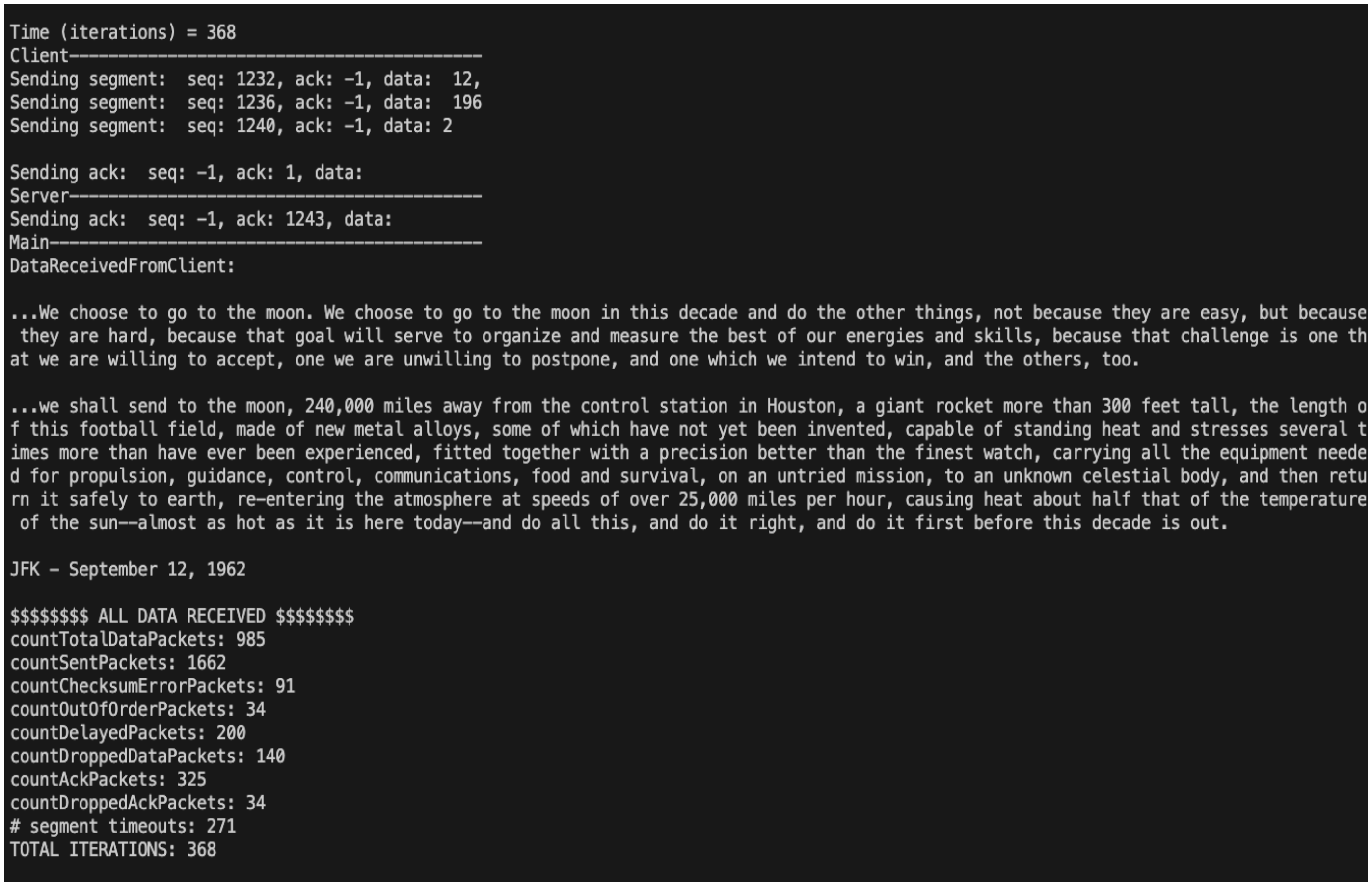
Screenshots:

Short text



```
Client------------------------------------------
Sending segment:  seq: 24, ack: -1, data: ed o
Sending segment:  seq: 28, ack: -1, data: ver
Sending segment:  seq: 32, ack: -1, data: the
Sending ack:  seq: -1, ack: 1, data:
Server------------------------------------------
Sending ack:  seq: -1, ack: 36, data:
Main--------------------------------------------
DataReceivedFromClient: The quick brown fox jumped over the
--------------------------------------------------------------------------------
Time (iterations) = 10
Client------------------------------------------
Sending segment:  seq: 36, ack: -1, data: lazy
Sending segment:  seq: 40, ack: -1, data:  dog
Sending ack:  seq: -1, ack: 1, data:
Server------------------------------------------
Sending ack:  seq: -1, ack: 40, data:
Main--------------------------------------------
DataReceivedFromClient: The quick brown fox jumped over the lazy dog
$$$$$$$$ ALL DATA RECEIVED $$$$$$$$
countTotalDataPackets: 27
countSentPackets: 44
countChecksumErrorPackets: 3
countOutOfOrderPackets: 1
countDelayedPackets: 2
countDroppedDataPackets: 3
countAckPackets: 10
countDroppedAckPackets: 1
# segment timeouts: 7
TOTAL ITERATIONS: 10
```

Large text



```
Time (iterations) = 368
Client------------------------------------------
Sending segment:  seq: 1232, ack: -1, data:  12,
Sending segment:  seq: 1236, ack: -1, data:  196
Sending segment:  seq: 1240, ack: -1, data: 2

Sending ack:  seq: -1, ack: 1, data:
Server------------------------------------------
Sending ack:  seq: -1, ack: 1243, data:
Main--------------------------------------------
DataReceivedFromClient:

...We choose to go to the moon. We choose to go to the moon in this decade and do the other things, not because they are easy, but because
 they are hard, because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one th
at we are willing to accept, one we are unwilling to postpone, and one which we intend to win, and the others, too.

...we shall send to the moon, 240,000 miles away from the control station in Houston, a giant rocket more than 300 feet tall, the length o
f this football field, made of new metal alloys, some of which have not yet been invented, capable of standing heat and stresses several t
imes more than have ever been experienced, fitted together with a precision better than the finest watch, carrying all the equipment neede
d for propulsion, guidance, control, communications, food and survival, on an untried mission, to an unknown celestial body, and then retu
rn it safely to earth, re-entering the atmosphere at speeds of over 25,000 miles per hour, causing heat about half that of the temperature
 of the sun--almost as hot as it is here today--and do all this, and do it right, and do it first before this decade is out.

JFK - September 12, 1962

$$$$$$$$ ALL DATA RECEIVED $$$$$$$$
countTotalDataPackets: 985
countSentPackets: 1662
countChecksumErrorPackets: 91
countOutOfOrderPackets: 34
countDelayedPackets: 200
countDroppedDataPackets: 140
countAckPackets: 325
countDroppedAckPackets: 34
# segment timeouts: 271
TOTAL ITERATIONS: 368
```