**9331 Assignment Report:**

**（python3）**

1. **Brief discussion of STP protocol:**

**Both in sender.py and receiver.py :**

Firstly, I create several classes including the function both in sender.py and receiver.py.

**STP\_Packet Class:**

This class is used for creating packet with head and data payload. When I send packet I create this class before and fill all the information and then send the packet.

**Sender class/Receiver class:**

These two class is pretty similar, both create UDP socket and also send data, receive data and so on, but Sender class is more complicated.

**sender.py:**

Sender Class:

* This class has the major function of the whole project. There are some basic parameters that requires to need like ip, port, file and so on. And also some other parameters that will be used in the process like current\_time, start\_time, and so on. And there is a socket creation in the class.
* There are lots of functions in it, like read the file, send packet, receive packet, retransmit packet and I combine the PLD in the class.
* Every packets that will be sent to the server will check the sender.PLD first, and return the different types of sign, which will use for implement different action in the way to transfer.

Return 1: Packet drop, Return 2: Packet duplicate; Return 3: Packet corrupt; Return 4: Packet reorder; Return 5: Packet delay; Return 0: Packet send successfully. In the function, it needs to check if the action is True or False, which is decided by the random number generator.

* Update\_log function is to record all the packet that has been sent or received.

**receiver.py:**

* Receiver class has some basic parameters and it creates a socket in the class. There are some functions similar to the class in sender.py like send\_data, recv\_data, and create typical type of packet, udate\_log and so on.
* It has get\_token function which is used for checksum.

**Main:**

* In the main part, I add some necessary variables for tracking state of the connection and some other variables to count the number of packet and check the time both in sender and receiver(the variables already get initialized).
* At first I start a while loop, so that it can always run the process:

While the connection is going to link, I create SYN, SYNACK, ACK, FIN packet to build a **3-way-hand-shake**, I tested it through the typical segement is true or not, so that the connection can be built.

* When the state variables has changed like the state\_established becomes true, it is ready to send and receive data and there is another wile loop to keep the process running(just for transfer real data part).

First I create window size, if size less than MWS//MSS it will be True.

Then I create the packet and start to send the data

* All the packet that is used to transfer data should check by the PLD\_check() function:

**Value 0:** it can transfer successfully.

**Value 1:** Packet drop happen and it writes in log and retransmit.

**Value 2:** Packet duplicate happen and it send twice, the first one send receive as usual but the second time it cannot send and receiver part will ignore it.

**Value 3:** Packet corrupt happen, it will change the bit and send to the receiver, receiver will send last time packet and the sender will send again. And in the receiver part, checksum in the packet will use md5 to check the data.

**Value 4:** Packet reorde happen, I save the packet in the re\_ordering\_list and increase the seq\_num and continue the next packet. But at the beginning of the loop(next packet begin) it needs to check if it already passes the maxOrder packet, and if True, just send the packet in the list.

**Value 5:** Packet delay happened, Unfortunately, when I code this part, there is always error happen so I just let it sleep(maxDely) time and retransmit.

* When the packet has been sent, it will start the timer(include initial timer, I use time.clock to track the timeout) and then if timeoutInterbal < time.clock() – timer, then it transmit over time so it needs to retransmit.

Then it will receive ACK packet from receiver and record in log.

By the way, I set some flag to deal with different situation.

* When the data\_progress is equal to the total data length, it send FIN packet to request stop connection and state will change, FIN-ACK, FIN-ACK.
* At last all the relative data like packet number will be recorded.

|  |  |
| --- | --- |
| **Implemented Features** | **Not Implemented** |
| 3 way hand shake | Delay process |
| 4 segment cnnt termination | Maximum window size(MWS) |
| Immediately acknowledgement | Timer retransmit |
| Timer operation |  |
| Drop retransmit |  |
| Duplicate process |  |
| Corrupt process |  |
| Reording process |  |
| Maximum Segment Size (MSS) |  |
| Sequence Num / Acknowledgement Num |  |

1. **STP Header:**

Duplicate process

Duplicate process

|  |
| --- |
| Sequence numbers |
| Acknowledgement numbers |
| ACK |
| SYN |
| FIN |
| PayLoad |
| Checksum |

The header includes these data:

Sequence numbers(int): give the sequence number of the packet

Acknowledgement numbers(int):

ACK(bool): Confirm whether it is valid

SYN(bool): Request to establish a connection

FIN(bool): Notify the connection close

Payload(MSS): data

Checksum: check data is valid

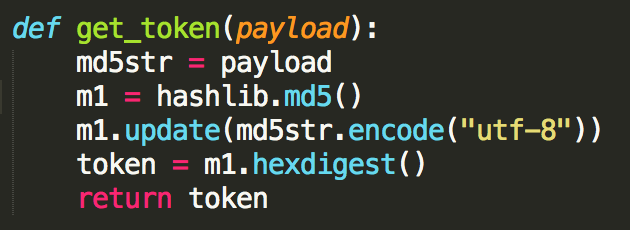
1. **Improvement:**

I can use Multithreading to decrease make the structure more clear.

I can improve the accuracy of packet transfer by using more head key segment.

Limit a top of transfer time and make the transfer faster.

1. **Borrow code:**

****

**5.**

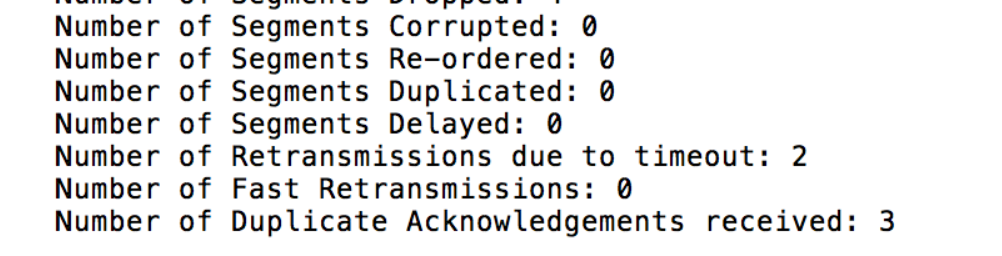
(a).

In the sample represents, the drop accured when the seq num is 301, 451 and 601.

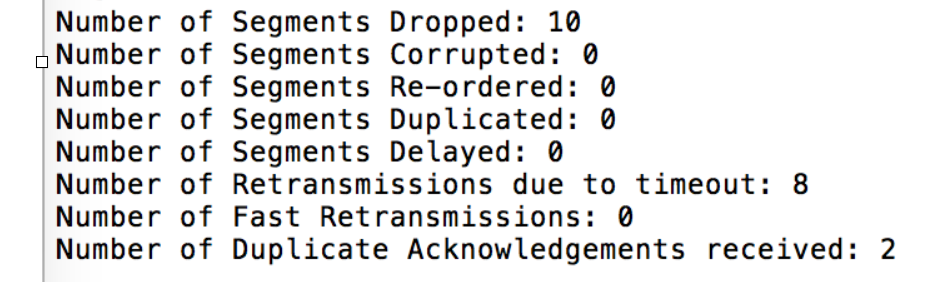
The number of drop is increase in pdrop = 0.3 because it has 30% > 10% to drop. So there are more drops in test2.

The test1 receives more duplicate ack. The drop segment will turn timeout when the max window size is full.

Test1:



Test2:



(b):

Sorry, I fail in the timeout part, and I don’t use the Multithreading, because I waste too much time on writing in single line, the process can just by single tread, and the timoutInterval cannot work as expect so I can’t answer this question. I already try my best, sorry.

(c):

The file can transferred successfully, but I’m sorry, just as I said before, I fail in timout and pdelay part, and some of the code may have problem, I have thought long time and correct or even rewrite my code, but it doesn’t work. It is so difficult for me, sorry I have tried my best.