CSC 361
Assignment 3

This assignment is done as the “modified P3” with both the client and server running on a terminal on the same IP address. The packet loss is done using random number instead of mininet.

The version of python used is python2.

RDP Header:

| | Possible Value | Meaning |
|---|---|---|
| Packet type (Name) | GET | Get request packet |
| | SYN | Synchronisation packet |
| | ACK | Acknowledgment packet |
| | FIN | Finish packet |
| | MD5 | MD5SUM packet |
| Sequence Number | | Packet sequence number |
| Acknowledgment Number | | Packet acknowledgement number |
| Payload | | Payload data |

**Working mechanism:**

The server and client are set running on the same IP address. The UDP sockets are used for both ends. The packets are created using strings and the different values are separated using commas (the commas in the file itself ae handled by combining the separated chunks of payload with commas).
The client sends a SYN packet to the server. The server responds back with an ACK. The client sends back another ACK. This completes our three-way handshake.

The client sends a GET request for a file from the server. This is taken from the command line argument. The server locates the file and sends it to the client by creating a list of words of size 1000 bytes from the file. The 1000 bytes serves as the payload length. The payload along with the name, sequence number and acknowledgment number form the header. The packets are sent continuously to the client till all the data is sent and received. After the sending is complete the server sends a FIN packet telling the client that it is the end of the transmission. The client sends back an ack telling the server that all the data has been received successfully.

The server takes the argument “TRUE” for the packet loss. If the argument does not equal “TRUE” then packet loss does not exist on the network.

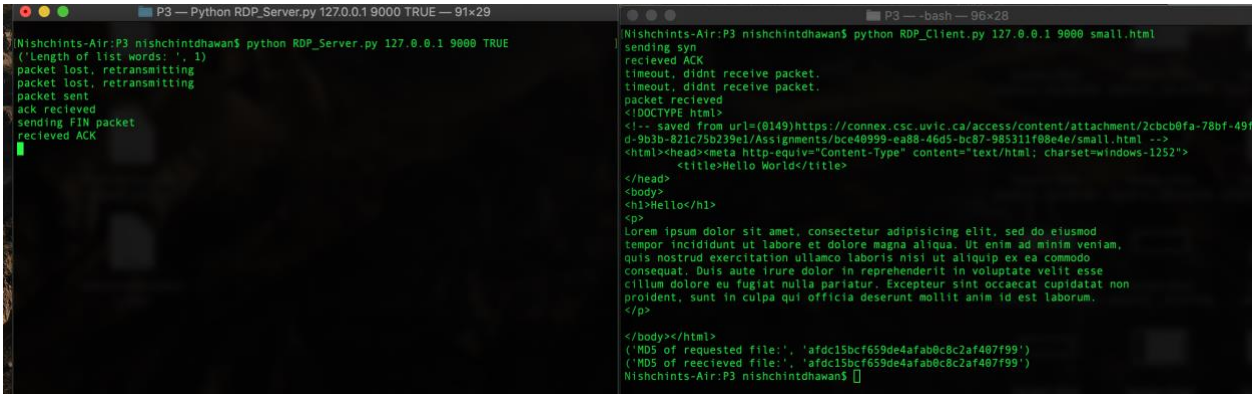
The client creates the file “received_file.txt” to store the server’s content.

**PACKET LOSS**:

The socket timeout is used to handle lost packets. On the server side, the packet is lost when the random number is less than 5. In such case, the server does not send a packet to the client but waits for the acknowledgment of that packet. As the client keeps waiting for the packet, it doesn’t send any acknowledgement back. The client and then the server times out which indicates the packet is lost. The server retransmits the packet.

**FILE CHECKING:**

The file is checked to be equal using the md5sum. Thee server sends the md5 sum before sending the file to the client. This is form of a md5 packet. After receiving the entire file, the client evaluates the md5 of its own and compares it with the md5 of the original file.
If the values are equal, then the contents are same. Hence, the data is successfully received.




The server side on the left takes in the argument TRUE for packet loss.

Link to the video :

https://drive.google.com/file/d/1-fieOhi87RmpEbL5noa7oU3ojPjj5Jrl/view?usp=sharing