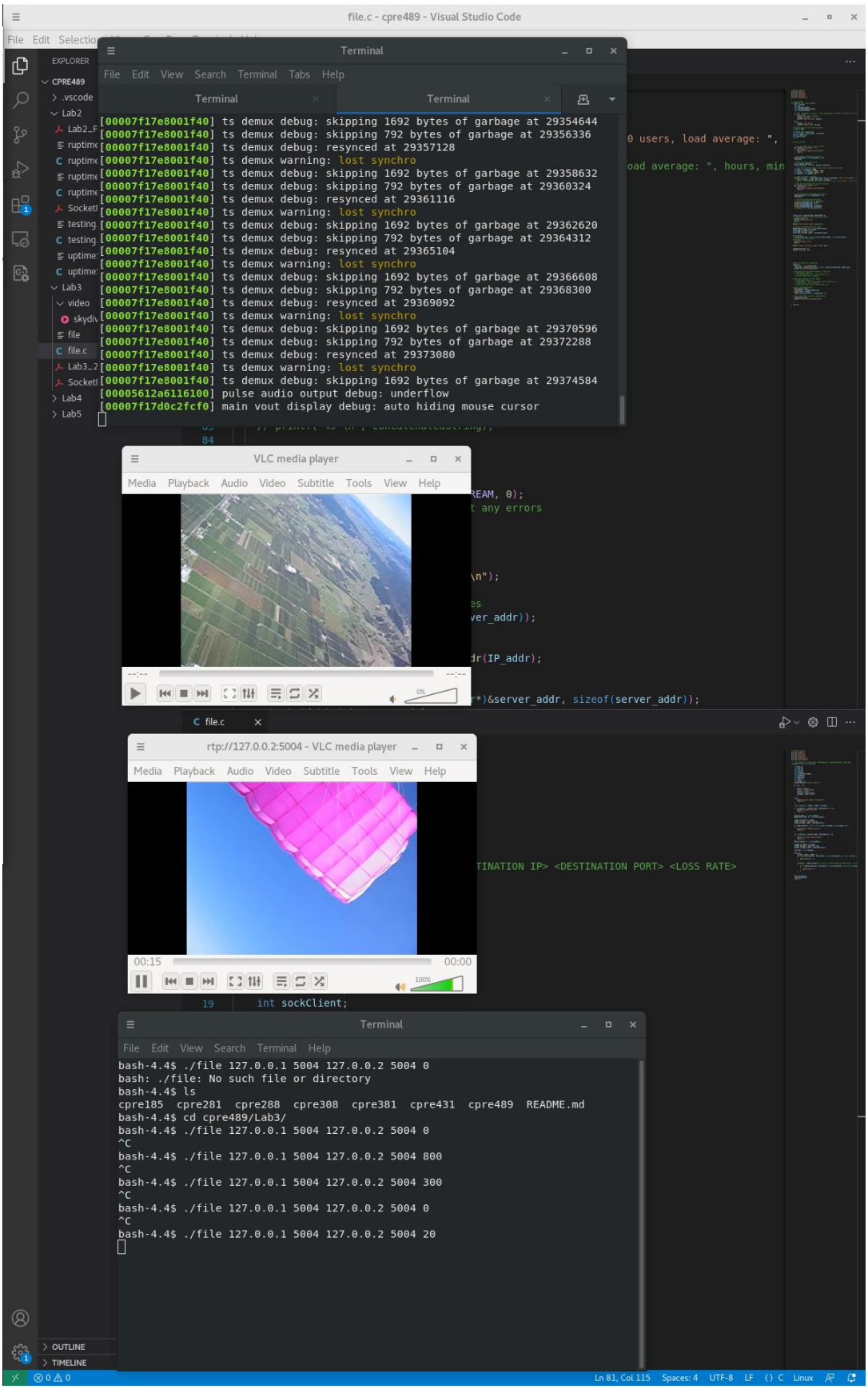
Lab 3

Summary

This lab teaches us about transferring data via the UDP protocol. As we know, UDP does not wait for the connection to be established before sending data. This makes it faster to use UDP for large files like a video file. In this lab, we send a video via VLC media player from one IP and port to another IP address. The C code we write routes the packets for this video file from the source and destination. It also

We play around with the loss rate values to see the quality of the video when the delay is increased and decreased.


file.c - cpre489 - Visual Studio Code
Terminal
[00007f17e8001f40] ts demux debug: skipping 1692 bytes of garbage at 29354644
[00007f17e8001f40] ts demux debug: skipping 792 bytes of garbage at 29356336
[00007f17e8001f40] ts demux debug: resynced at 29357128
[00007f17e8001f40] ts demux warning: lost synchro
[00007f17e8001f40] ts demux debug: skipping 1692 bytes of garbage at 29358632
[00007f17e8001f40] ts demux debug: skipping 792 bytes of garbage at 29360324
[00007f17e8001f40] ts demux debug: resynced at 29361116
[00007f17e8001f40] ts demux warning: lost synchro
[00007f17e8001f40] ts demux debug: skipping 1692 bytes of garbage at 29362620
[00007f17e8001f40] ts demux debug: skipping 792 bytes of garbage at 29364312
[00007f17e8001f40] ts demux debug: resynced at 29365104
[00007f17e8001f40] ts demux warning: lost synchro
[00007f17e8001f40] ts demux debug: skipping 1692 bytes of garbage at 29366608
[00007f17e8001f40] ts demux debug: skipping 792 bytes of garbage at 29368300
[00007f17e8001f40] ts demux debug: resynced at 29369092
[00007f17e8001f40] ts demux warning: lost synchro
[00007f17e8001f40] ts demux debug: skipping 1692 bytes of garbage at 29370596
[00007f17e8001f40] ts demux debug: skipping 792 bytes of garbage at 29372288
[00007f17e8001f40] ts demux debug: resynced at 29373080
[00007f17e8001f40] ts demux warning: lost synchro
[00007f17e8001f40] ts demux debug: skipping 1692 bytes of garbage at 29374584
[00005612a6116100] pulse audio output debug: underflow
[00007f17d0c2fcf0] main vout display debug: auto hiding mouse cursor
VLC media player
rtp://127.0.0.2:5004 - VLC media player
Terminal
bash-4.4$ ./file 127.0.0.1 5004 127.0.0.2 5004 0
bash: ./file: No such file or directory
bash-4.4$ ls
cpre185 cpre281 cpre288 cpre308 cpre381 cpre431 cpre489 README.md
bash-4.4$ cd cpre489/Lab3/
bash-4.4$ ./file 127.0.0.1 5004 127.0.0.2 5004 0
^C
bash-4.4$ ./file 127.0.0.1 5004 127.0.0.2 5004 800
^C
bash-4.4$ ./file 127.0.0.1 5004 127.0.0.2 5004 300
^C
bash-4.4$ ./file 127.0.0.1 5004 127.0.0.2 5004 0
^C
bash-4.4$ ./file 127.0.0.1 5004 127.0.0.2 5004 20

Exercises

Exercise 1

Increasing the loss rate diminishes the quality of the received video. We can see a lot of pixelated frames within the video and a huge delay for it to start playing on the receiver end. Even when the loss rate is set to 1, we see a considerable delay. When it's 0, the quality is much better, and the delay is minute.

Exercise 2

In case of a pre-existing local file that needs to be propagated within a local network, TCP would be a good choice since the average delay will be much less. But in the case of propagating a video from the internet or some far-off server, TCP would not be a good choice. This is because TCP works on the principle of handshaking and only starts propagating packets once the established connection is confirmed and the ACKs are sent back from the receiver to the sender. UDP does not wait for a confirmation of the established connection. This makes the delay less.