

**Ahmedabad  
University**

## Computer Networks

**Group No. 17**

Submitted to faculty: Shashi Prabh

Date of Submission: 9th May

### Student Details

Roll No.	Name of the Student	Name of the Program
AU1940151	Purvam Sheth	B-Tech CSE
AU1940261	Krutin Rathod	B-Tech CSE
AU1940215	Raj Chauhan	B-Tech CSE
AU1940119	Nihar Patel	B-Tech CSE
AU1940171	Mohit Prajapati	B-Tech CSE

## Python files :

cn\_proj contains two files for client and server.

server\_code\_grp\_17.py : Multicasting video streaming code of server side i.e. sender.

client\_code\_grp\_17.py : Multicasting video streaming code of client side i.e. receiver.

Video folder contains videos to be played by our application.

## How to execute our code :

1. First we will run the server side code by passing the following arguments : ip address of the device, multicasting address no .1 and corresponding port no. 1 to use, then second multicasting address and corresponding port no. 2, then we have to pass the message to be sent (if any), and lastly port no. of TCP client.
2. Then we will run the client side code by passing the following command line arguments : ip address of the device, port no. to use in order to connect with the TCP client.
3. In short, the client will send a request to the server to join the multicast group.
4. Then the server will provide a list of station info to the client through TCP connection.
5. Then the selected station from the station list is connected to that station.
6. And the specific video related to that station will be played using the opencv library.
7. After the video starts playing, the client will have the option to play, pause, resume and terminate the video at any given time.
8. By pressing “p” we can pause the video played on the client side.
9. By pressing “r” the video will start playing from the location where the transmitting video is being played i.e. just like radio stations.
10. By pressing “t” the video will terminate and exit out of the application.

## Explanation:

### **Functions of server\_code::**

Helper\_function) - this will give an error if the command line arguments are not enough as required.

Multicast\_send - This function will consist of all the steps that the server will have to perform in order to connect with the clients. First step is to initialize the socket , second command will define how many hops a multicast datagram can travel. Next step is to define the network

interface which is responsible for transmitting the multicast datagram. Last steps are to send the datagrams in the buffer to the clients and close the socket.

**video\_Multicasting**- This function takes input of the host ip addresses , multicast group ip addresses, multicast port, and path to the videos. Firstly, we will start capturing initialized video using open cv2. Then we will run those videos in loop forever while displaying the FPS of the video.

**Tcp** - This function will implement all the TCP sequences like creating a socket, binding, listening and accepting to the clients. We created 2 stations from and we sent both of them via send function.

**Main** - Firstly we will check if the arguments in the are up to the mark or not by using the helper function. Then we will assign all the variables that will be used that are recovered from argument commands. Another thing that we had to implement was multi-processing to display multiple videos together simultaneously. So we created two processes, one for each video we had to call the function of `print_time` here by giving all the arguments the user entered through the command prompt and then starting the processes.

#### **Functions of client\_code:**

All of the functions inherit same functionalities as the sender file, but from the receiver's perspective. For example in `Multicast_receive`, instead of sending data we receive the data from the server.

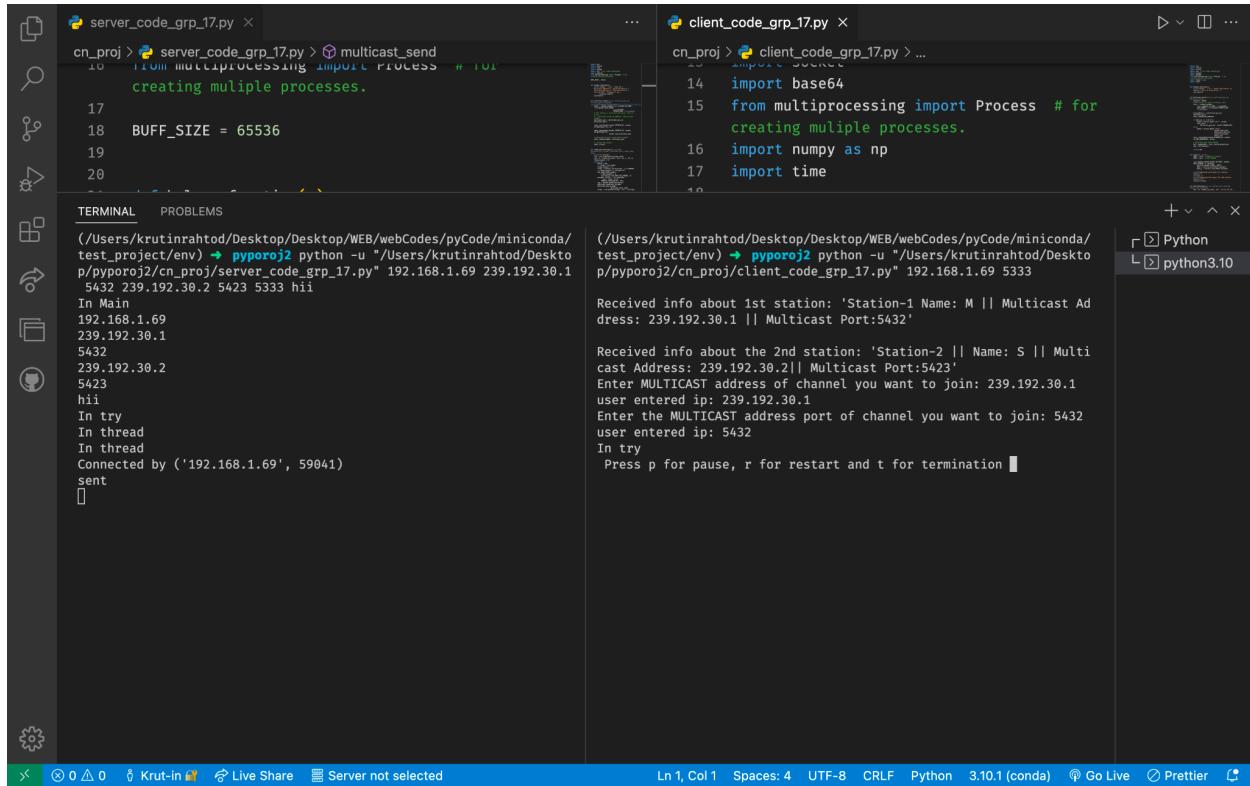
## Functionality

1. **Pause** : pressing p key during transmission and receiving of the video will close the client screen and he won't be able to see anything.
2. **Restart**: pressing the r key after pausing the video will cause a window to pop up and resume the video from where transmitting video is being played, of course if compared to radio stations, if we don't listen to the radio, the radio keeps on going forward.
3. **Terminate**: close the video and exit out of the application.

**Features:** The transmitting video will play on loop and thus going on forever. Used TCP connection to send messages and UDP for sending multimedia files.

## Screenshots

Image 1 shows the command line arguments required by the client as well as server side code. We can also see the various functionalities being implemented as discussed above. The client has to select one of the given stations to be played on his side i.e. the receiving side. The corresponding multicasting address and port no. are selected. Thus connection is established between them.



The screenshot shows a code editor with two terminal windows. The left terminal window displays the server code (server\_code\_grp\_17.py) which creates multiple processes and sends data to a multicast address. The right terminal window displays the client code (client\_code\_grp\_17.py) which receives info about stations, enters a multicast address, and starts playing a video. The code uses Python's multiprocessing module and numpy for playing video.

```
server_code_grp_17.py
...
17
18 BUFF_SIZE = 65536
19
20

client_code_grp_17.py
...
14 import base64
15 from multiprocessing import Process # for
16 creating multiple processes.
17 import numpy as np
18
19
20

TERMINAL PROBLEMS
(/Users/krutinrahtod/Desktop/Desktop/WEB/webCodes/pyCode/miniconda/test_project/env) ➔ pyporo2 python -u "/Users/krutinrahtod/Desktop/pyporo2/cn_proj/server_code_grp_17.py" 192.168.1.69 239.192.30.1
5432 239.192.30.2 5423 5333 hii
In Main
192.168.1.69
239.192.30.1
5432
239.192.30.2
5423
hii
In try
In thread
In thread
Connected by ('192.168.1.69', 59041)
sent
[]

(/Users/krutinrahtod/Desktop/Desktop/WEB/webCodes/pyCode/miniconda/test_project/env) ➔ pyporo2 python -u "/Users/krutinrahtod/Desktop/pyporo2/cn_proj/client_code_grp_17.py" 192.168.1.69 5333
Received info about 1st station: 'Station-1 Name: M || Multicast Address: 239.192.30.1 || Multicast Port:5432'
Received info about the 2nd station: 'Station-2 || Name: S || Multicast Address: 239.192.30.2|| Multicast Port:5423'
Enter MULTICAST address of channel you want to join: 239.192.30.1
user entered ip: 239.192.30.1
Enter the MULTICAST address port of channel you want to join: 5432
user entered ip: 5432
In try
Press p for pause, r for restart and t for termination
```

Image 2 shows 3 videos showing. From those 3, 2 are transmitting videos, and the 3rd one is chosen by the client in order to stream. On that one client has functionalities to pause, play and terminate the video.

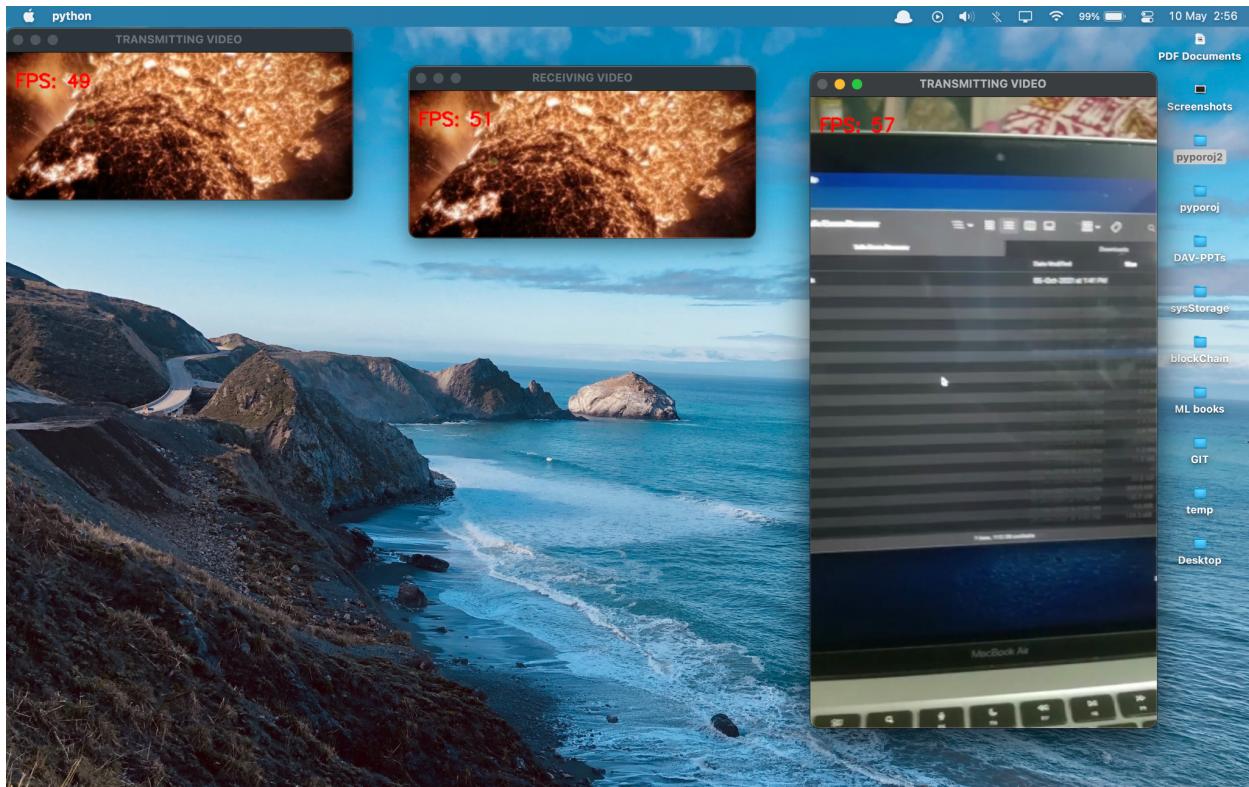


Image 3 shows the functionalities in action and corresponding messages.

```

server_code_grp_17.py
cn_proj > server_code_grp_17.py > multicast_send
10  from multiprocessing import Process # for
     creating multiple processes.
11
12  BUFF_SIZE = 65536
13
14
15
16
17
18
19
20

client_code_grp_17.py
cn_proj > client_code_grp_17.py > ...
10  import socket
11
12  import base64
13  from multiprocessing import Process # for
     creating multiple processes.
14  import numpy as np
15  import time
16
17
18
19
20

TERMINAL
(/Users/krutinrahtod/Desktop/Desktop/WEB/webCodes/pyCode/miniconda/test_project/env) ➔ pyporoj2 python -u "/Users/krutinrahtod/Desktop/pyporoj2/cn_proj/server_code_grp_17.py" 192.168.1.69 239.192.30.1
5432 239.192.30.2 5423 5333 hii
In Main
192.168.1.69
239.192.30.1
5432
239.192.30.2
5423
hii
In try
In thread
In thread
Connected by ('192.168.1.69', 59041)
sent
[]

(/Users/krutinrahtod/Desktop/Desktop/WEB/webCodes/pyCode/miniconda/test_project/env) ➔ pyporoj2 python -u "/Users/krutinrahtod/Desktop/pyporoj2/cn_proj/client_code_grp_17.py" 192.168.1.69 5333
Received info about 1st station: 'Station-1 Name: M || Multicast Address: 239.192.30.1 || Multicast Port:5432'
Received info about the 2nd station: 'Station-2 || Name: S || Multicast Address: 239.192.30.2|| Multicast Port:5423'
Enter MULTICAST address of channel you want to join: 239.192.30.1
user entered ip: 239.192.30.1
Enter the MULTICAST address port of channel you want to join: 5432
user entered ip: 5432
In try
Press p for pause, r for restart and t for termination p
Pause clicked:
Press p for pause, r for restart and t for termination r
r pressed:
Press p for pause, r for restart and t for termination t
t pressed
(/Users/krutinrahtod/Desktop/Desktop/WEB/webCodes/pyCode/miniconda/test_project/env) ➔ pyporoj2

```

Image 4 shows the view after the client terminates his video. Hence only videos transmitted by the server can be seen.

