Computer Networks
Project Report

Group number : 5

1) Dhara Vora	: 1741046
2) Kalagee Anjaria	: 1741052
3) Nidhi Golakiya	: 1741062
4) Kruti Yadav	: 1741096

Objective :

The main objective of our project is to develop a radio system live streaming using socket programming.

We are developing an Internet Radio application that uses multicast. In our project we have developed Any Source Multicast (ASM) model. Multicast messages are identified by the multicast group address alone, and more than one sender can exist in a group. Therefore, any message sent to a multicast address is received by all nodes that have joined the group.

Implementation :

We have developed one client, one server, three senders and one receiver. The three senders continuously keep sending data for the **three stations** we have created. 1st station is for english songs, 2nd for hindi songs, and 3rd for gujarati songs. Then, the server sends a list of stations to the client after connection is established. As soon as the client selects to go on any one station, it calls the receiver and the receiver starts receiving data from that respective sender. As we can multicast, different clients can opt for different stations at the same time, and the corresponding sender sends that data to the receiver connected.

The client establishes a control channel using TCP socket with port number 5430 and receives multimedia streams over a UDP socket based on the port number of senders. Different stations(senders) have different port numbers. (Sender1 : 5431, Sender2 : 5432 , Sender3 : 5433). We have used **239.192.5.10** as the multicast address.

Then, to make our project more presentable, we have also applied **Graphical User Interface(GUI)**. As soon as the client is connected, the **1st window** is displayed which contains four buttons, each for the **three stations** and one **terminate button** respectively. Then, when the client clicks on the station button he wants to visits, the receiver is called and that sender starts sending data for that station. And, **another window appears**, which again contains for buttons : **Quit, Pause, Resume and Change station**. The receiver can quit receiving the data by clicking on the 1st button, pause the streaming by 2nd, resume the streaming again by 3rd and choose to leave that station and select another one by 4th. While this window appears, the previous one still remains, only it is disabled till the time the client again needs to choose a station or wish to terminate it.

- In the receiver file, GUI has an infinite loop to keep displaying it and another infinite loop to keep receiving data from sender. Both can’t work together so we have used **thread to receive data and run GUI simultaneously**.

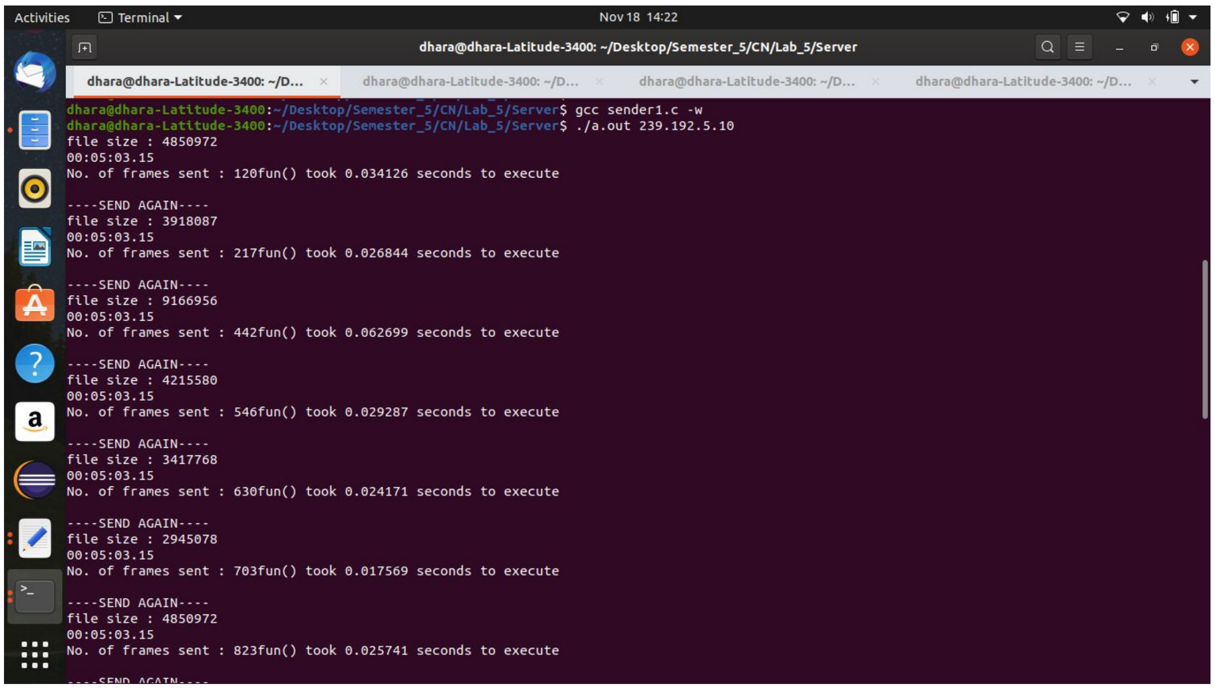
- For, control functions like quit,pause and resume , we have used thread commands like - **kill, stop and cont(continue)** respectively.
    - To use these commands we need **process id(PID)**. For that we have extracted the process id using ‘awk’ command from ‘ps-ef’. **ps command** is used to list the currently running processes and their PIDs along with some other information depends on different options.

Bit rate and buffer size calculation :

We have converted the audio files to streamable audio file using ‘**ffmpeg -i input_file_name -f mpegts output_file_name’** command before sending it to the receiver. By doing this we are getting the same bit rate of each audio file. We have defined t=7 seconds in the receiver file and according to the bit rate and time t, we have calculated **buffer size**(In our case it is **64K**).

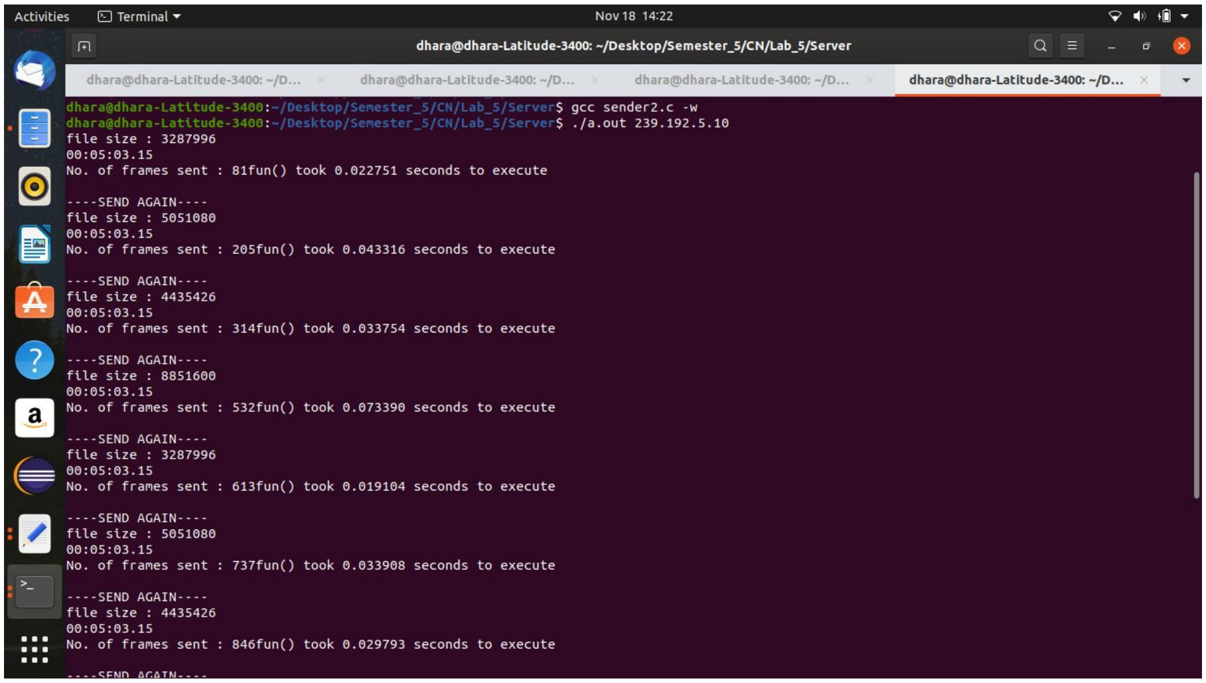
Screenshots :

1) Sender1 is continuously sending the data in a loop.


Activities
Terminal
Nov 18 14:22
dhara@dhara-Latitude-3400: ~/Desktop/Semester_5/CN/Lab_5/Server
dhara@dhara-Latitude-3400: ~/D...
dhara@dhara-Latitude-3400: ~/D...
dhara@dhara-Latitude-3400: ~/D...
dhara@dhara-Latitude-3400: ~/D...

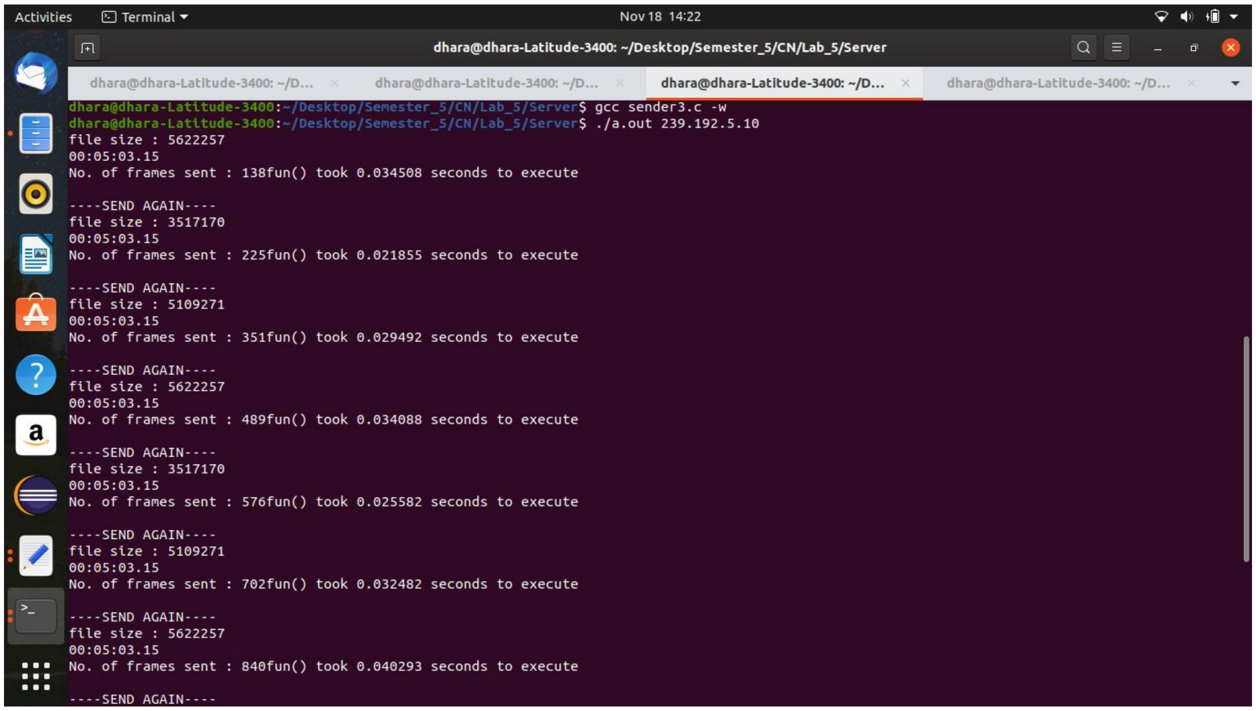
```
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ gcc sender1.c -w
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ ./a.out 239.192.5.10
file size : 4850972
00:05:03.15
No. of frames sent : 120fun() took 0.034126 seconds to execute

----SEND AGAIN----
file size : 3918087
00:05:03.15
No. of frames sent : 217fun() took 0.026844 seconds to execute

----SEND AGAIN----
file size : 9166956
00:05:03.15
No. of frames sent : 442fun() took 0.062699 seconds to execute

----SEND AGAIN----
file size : 4215580
00:05:03.15
No. of frames sent : 546fun() took 0.029287 seconds to execute

----SEND AGAIN----
file size : 3417768
00:05:03.15
No. of frames sent : 630fun() took 0.024171 seconds to execute

----SEND AGAIN----
file size : 2945078
00:05:03.15
No. of frames sent : 703fun() took 0.017569 seconds to execute

----SEND AGAIN----
file size : 4850972
00:05:03.15
No. of frames sent : 823fun() took 0.025741 seconds to execute

----SEND AGAIN----
```

2) Sender2 is continuously sending the data in a loop.



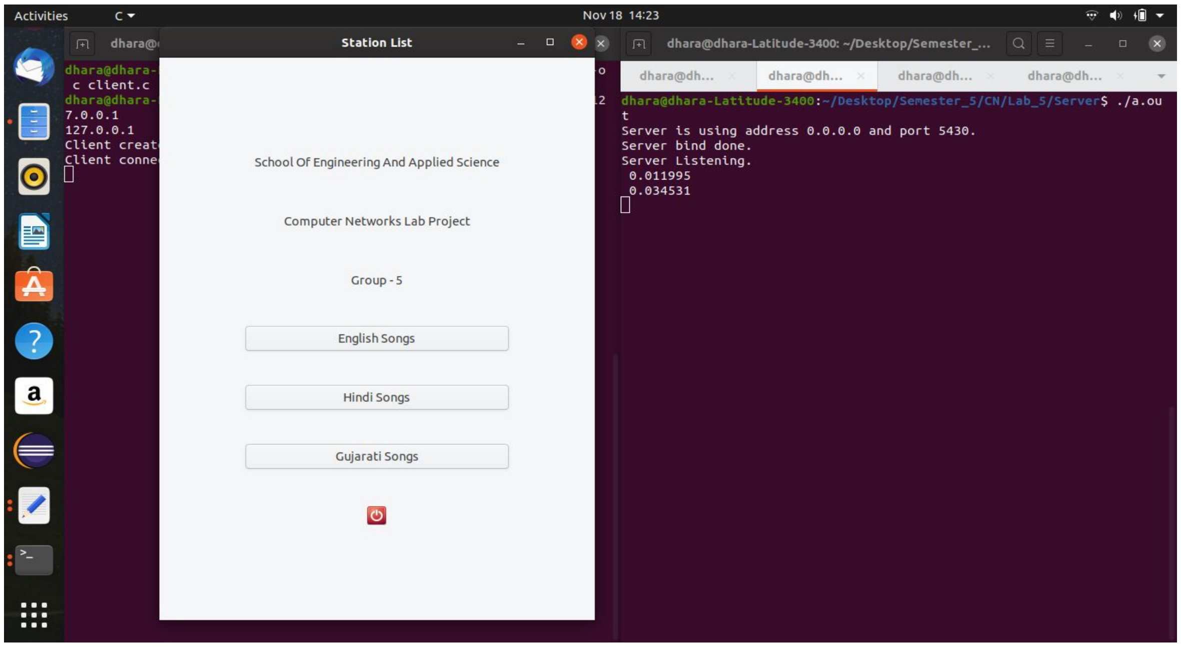
```
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ gcc sender2.c -w
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ ./a.out 239.192.5.10
file size : 3287996
00:05:03.15
No. of frames sent : 81fun() took 0.022751 seconds to execute

----SEND AGAIN----
file size : 5051080
00:05:03.15
No. of frames sent : 205fun() took 0.043316 seconds to execute

----SEND AGAIN----
file size : 4435426
00:05:03.15
No. of frames sent : 314fun() took 0.033754 seconds to execute

----SEND AGAIN----
file size : 8851600
00:05:03.15
No. of frames sent : 532fun() took 0.073390 seconds to execute

----SEND AGAIN----
file size : 3287996
00:05:03.15
No. of frames sent : 613fun() took 0.019104 seconds to execute

----SEND AGAIN----
file size : 5051080
00:05:03.15
No. of frames sent : 737fun() took 0.033908 seconds to execute

----SEND AGAIN----
file size : 4435426
00:05:03.15
No. of frames sent : 846fun() took 0.029793 seconds to execute

----SEND AGAIN----
```

3) Sender3 is continuously sending the data in a loop.



```
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ gcc sender3.c -w
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ ./a.out 239.192.5.10
file size : 5622257
00:05:03.15
No. of frames sent : 138fun() took 0.034508 seconds to execute

----SEND AGAIN----
file size : 3517170
00:05:03.15
No. of frames sent : 225fun() took 0.021855 seconds to execute

----SEND AGAIN----
file size : 5109271
00:05:03.15
No. of frames sent : 351fun() took 0.029492 seconds to execute

----SEND AGAIN----
file size : 5622257
00:05:03.15
No. of frames sent : 489fun() took 0.034088 seconds to execute

----SEND AGAIN----
file size : 3517170
00:05:03.15
No. of frames sent : 576fun() took 0.025582 seconds to execute

----SEND AGAIN----
file size : 5109271
00:05:03.15
No. of frames sent : 702fun() took 0.032482 seconds to execute

----SEND AGAIN----
file size : 5622257
00:05:03.15
No. of frames sent : 840fun() took 0.040293 seconds to execute

----SEND AGAIN----
```

4) This GUI window opens when the client runs and is connected to the server.The first three buttons are for the three stations and fourth one is for termination.

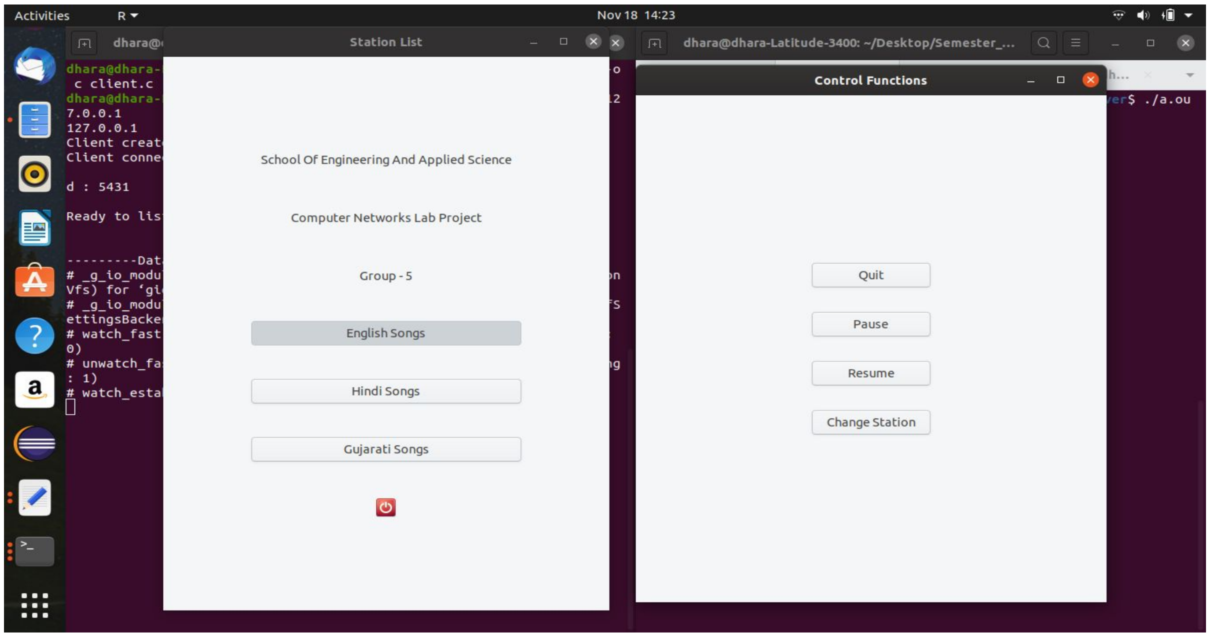


```
dhara@dhara-
 c client.c
dhara@dhara-
7.0.0.1
127.0.0.1
Client creat
Client conne
```

Station List
School Of Engineering And Applied Science
Computer Networks Lab Project
Group - 5
English Songs
Hindi Songs
Gujarati Songs

```
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ ./a.ou
t
Server is using address 0.0.0.0 and port 5430.
Server bind done.
Server Listening.
 0.011995
 0.034531
```

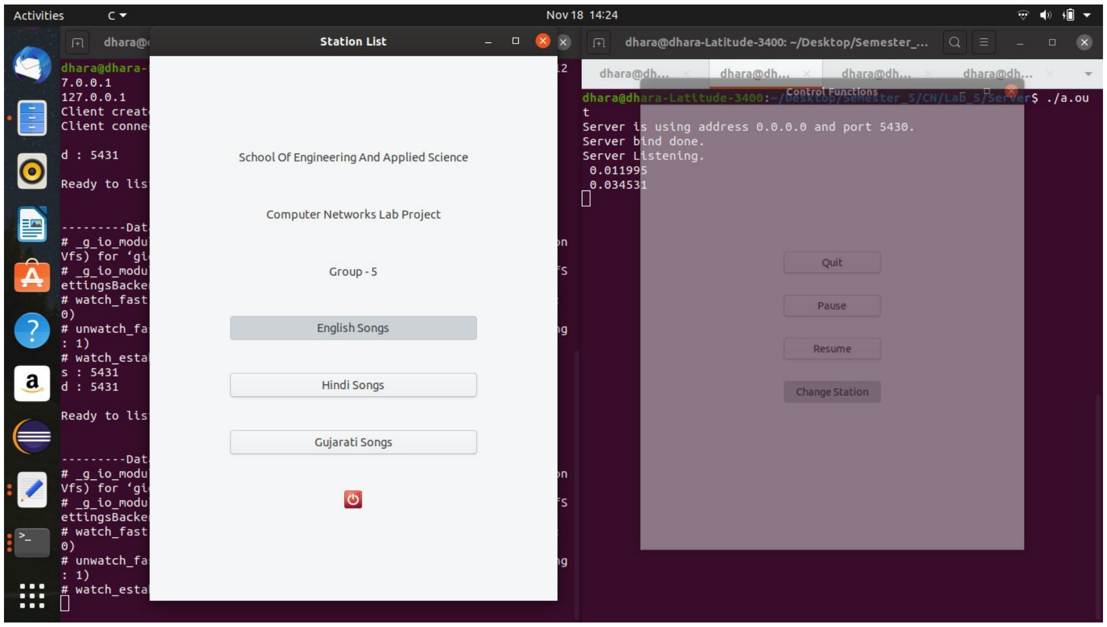
5) When the client selects let’s say the station for english songs, another window consisting of the control functions opens.



```
dhara@dhara-
 c client.c
dhara@dhara-
7.0.0.1
127.0.0.1
Client creat
Client conne

d : 5431

Ready to lis

---------Dat
# _g_io_modu
Vfs) for 'gi
# _g_io_modu
ettingsBacke
# watch_fast
0)
# unwatch_fa
: 1)
# watch_esta
```

```
er$ ./a.ou
```

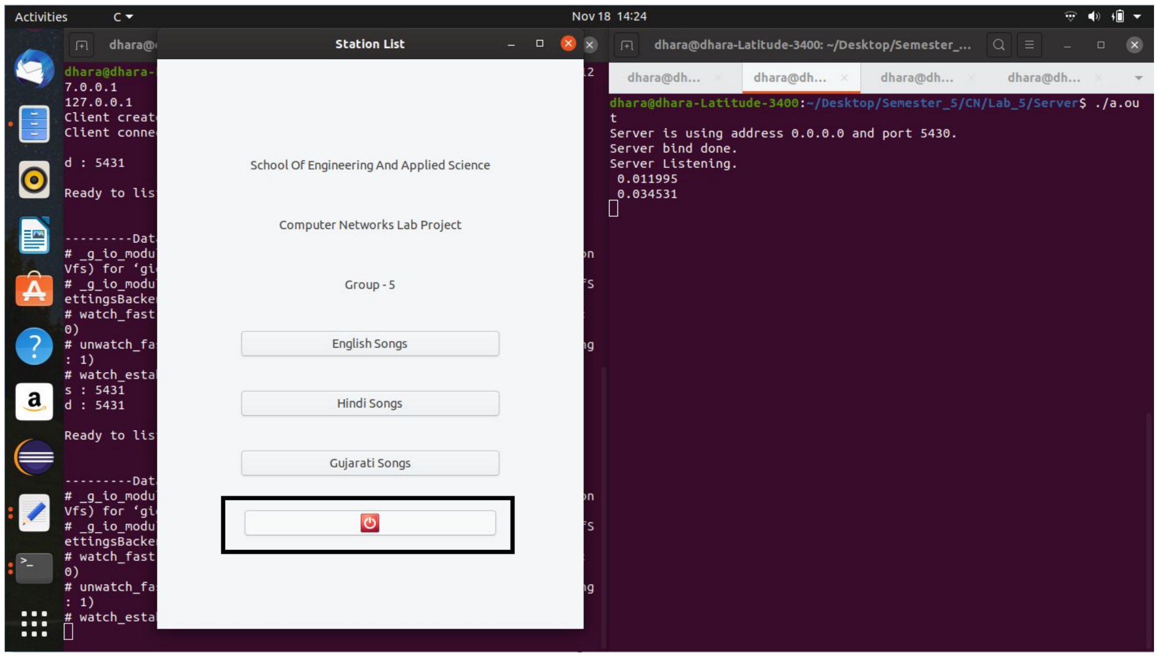
6) This shows that when the receiver chooses change station, the control functions window closes and we can again choose the desired station from the previous window.



```
dhara@dhara-
7.0.0.1
127.0.0.1
Client creat
Client conne

d : 5431

Ready to lis

---------Dat
# _g_io_modu
Vfs) for 'gi
# _g_io_modu
ettingsBacke
# watch_fast
0)
# unwatch_fa
: 1)
# watch_esta
s : 5431
d : 5431

Ready to lis

---------Dat
# _g_io_modu
Vfs) for 'gi
# _g_io_modu
ettingsBacke
# watch_fast
0)
# unwatch_fa
: 1)
# watch_esta
```

```
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ ./a.ou
t
Server is using address 0.0.0.0 and port 5430.
Server bind done.
Server Listening.
 0.011995
 0.034531
```

7) After that, finally, if the client wants to terminate the application itself, it clicks on the fourth button highlighted in the image below.



```
dhara@dhara-
7.0.0.1
127.0.0.1
Client creat
Client conne

d : 5431

Ready to lis


---------Dat
# _g_io_modu
Vfs) for ‘gi
# _g_io_modu
ettingsBacke
# watch_fast
0)
# unwatch_fa
: 1)
# watch_esta
s : 5431
d : 5431

Ready to lis


---------Dat
# _g_io_modu
Vfs) for ‘gi
# _g_io_modu
ettingsBacke
# watch_fast
0)
# unwatch_fa
: 1)
# watch_esta
```

```
dhara@dhara-Latitude-3400:~/Desktop/Semester_5/CN/Lab_5/Server$ ./a.ou
t
Server is using address 0.0.0.0 and port 5430.
Server bind done.
Server Listening.
 0.011995
 0.034531
```

Contributions :

Sender, Receiver Code	: Nidhi, Kruti
Server, Client Code	: Dhara, Kalagee
GUI Design	: Kruti, Kalagee
GUI connection with socket programming	: Nidhi, Dhara
Report	: Dhara, Nidhi, Kalagee, Kruti

Demo video link :

https://drive.google.com/open?id=1FzddoL2KxqHJASJreR1MhMSmHYo_oube