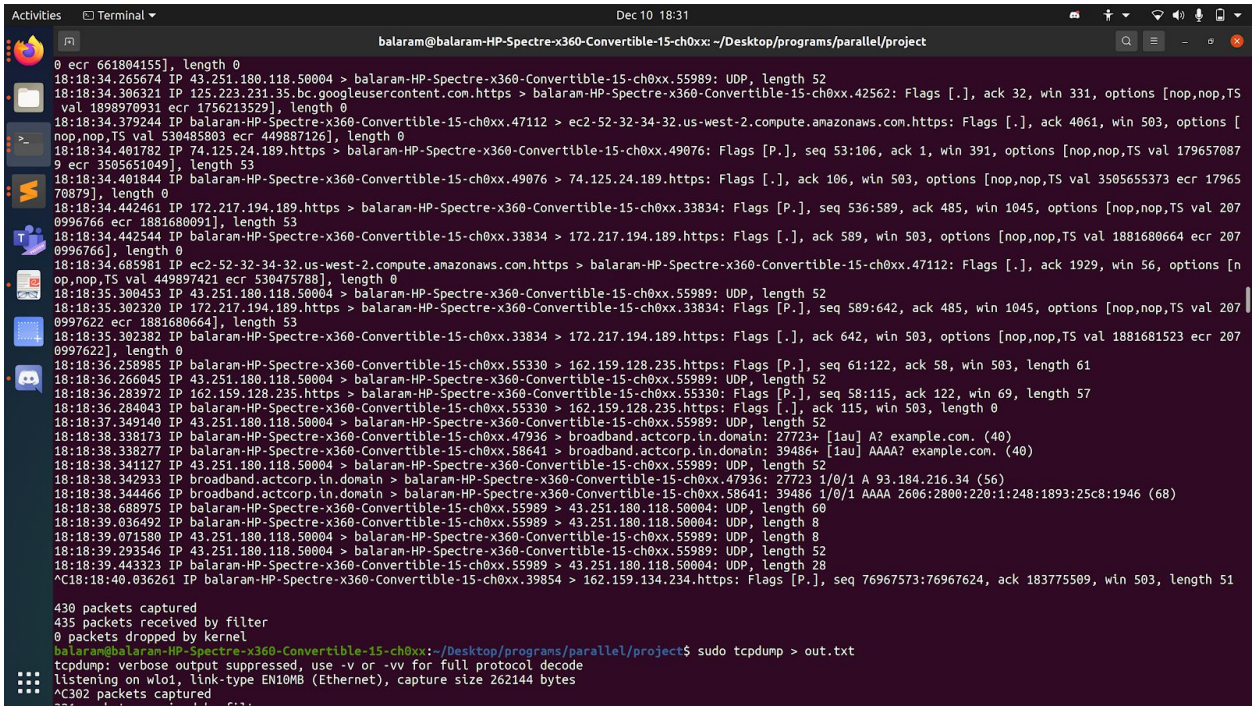
CS3530-Computer Networks-1

Team Members:
Sai Balaram K- ES18BTECH11011
P V Asish- CS18BTECH11037
Sai Nishith J- CS18BTECH11018
G V Sathwik Reddy- CS18BTECH11014
P Sai Srikar- CS18BTECH11034
B Sai Laxman- EE18BTECH11049

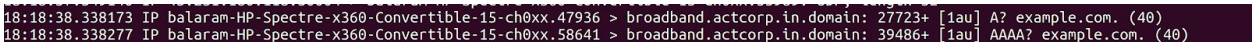
Design

1) getaddrinfo() integrated in code to get the ip from the host name passed as the command line argument. Can see the pic attached below of tcpdump

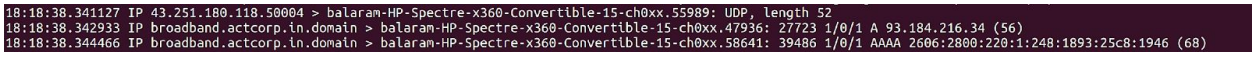
tcpdump collected on wlo1 while program for getaddrinfo() request was sent on another terminal



U can see the magnified image below here sending a DNS query about the host name through the internet which is going to my ISP.



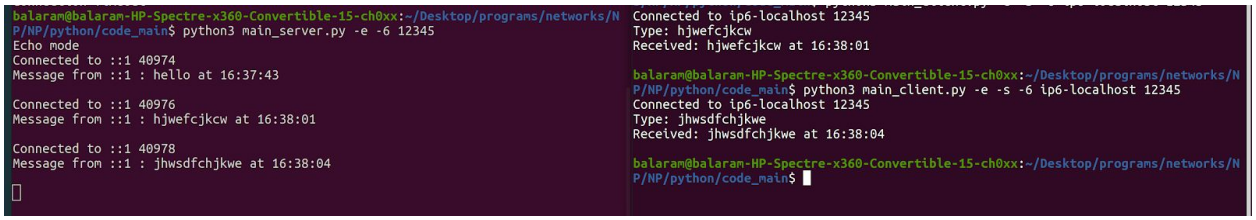
I get back the answer to the query which you can see in the 2nd image below.



We can see that we get back the ip address related to the query both ipv4 and ipv6

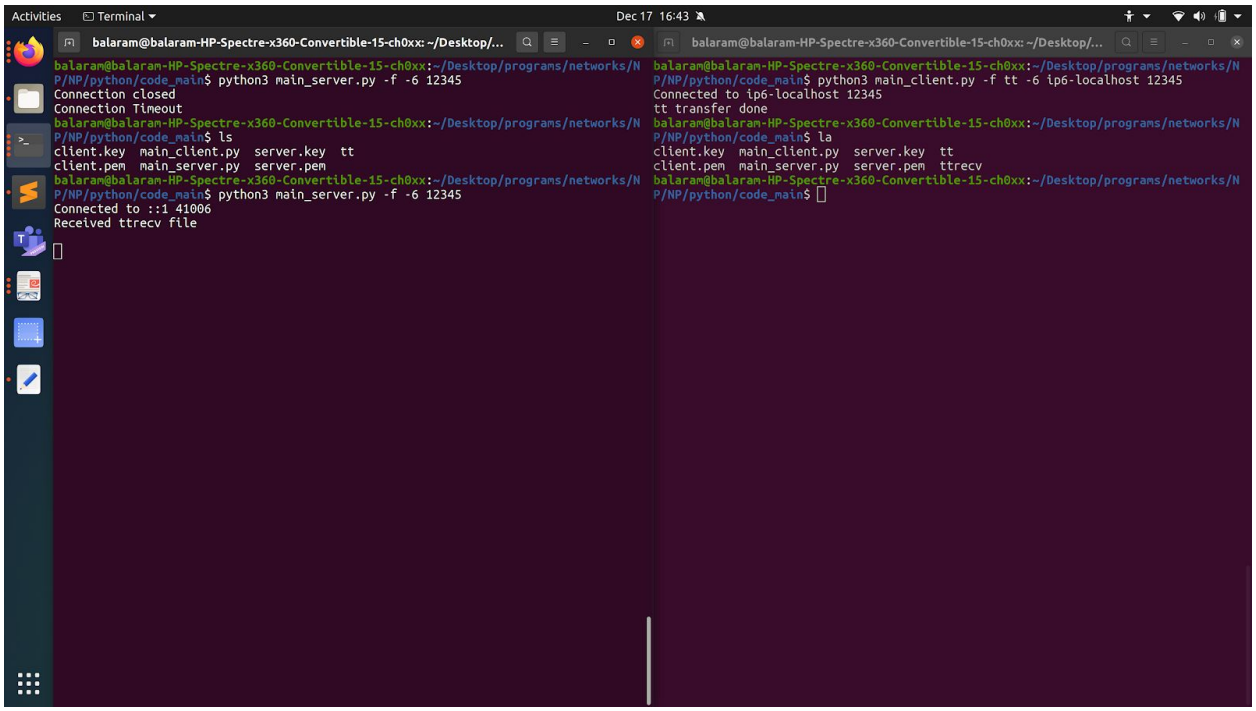
2) Other features added in the code provided are:

i) adding timestamp while sending the code in echo mode which we can know when we sent the message when we get back the echo client:



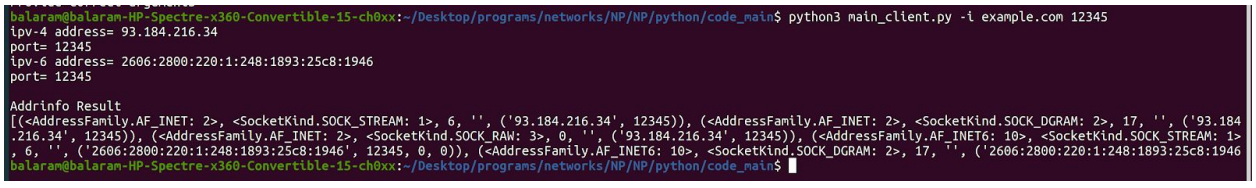
U can observer the time stamp mentioned in the echo of the client as well as the server which receives the echo

ii)file transfer using sockets can be useful in file transfer through socket programming which will be a very useful transfer of file from one to another place.



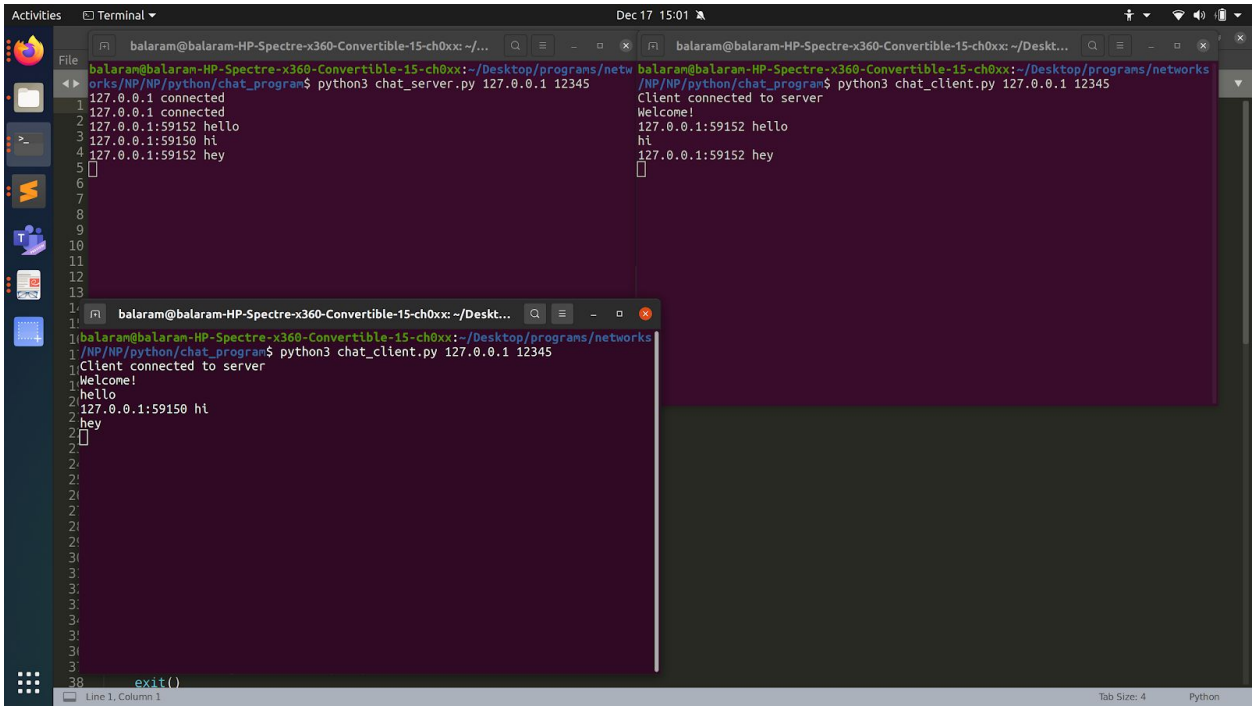
U can see in the image before and after ls command the file transfer done, added recv at end of recv to know explicit way that file is received

iii) getaddrinfo in the client program to get the information according to the hostname instead of passing the ip address. Get addrinfo integrated on the client side



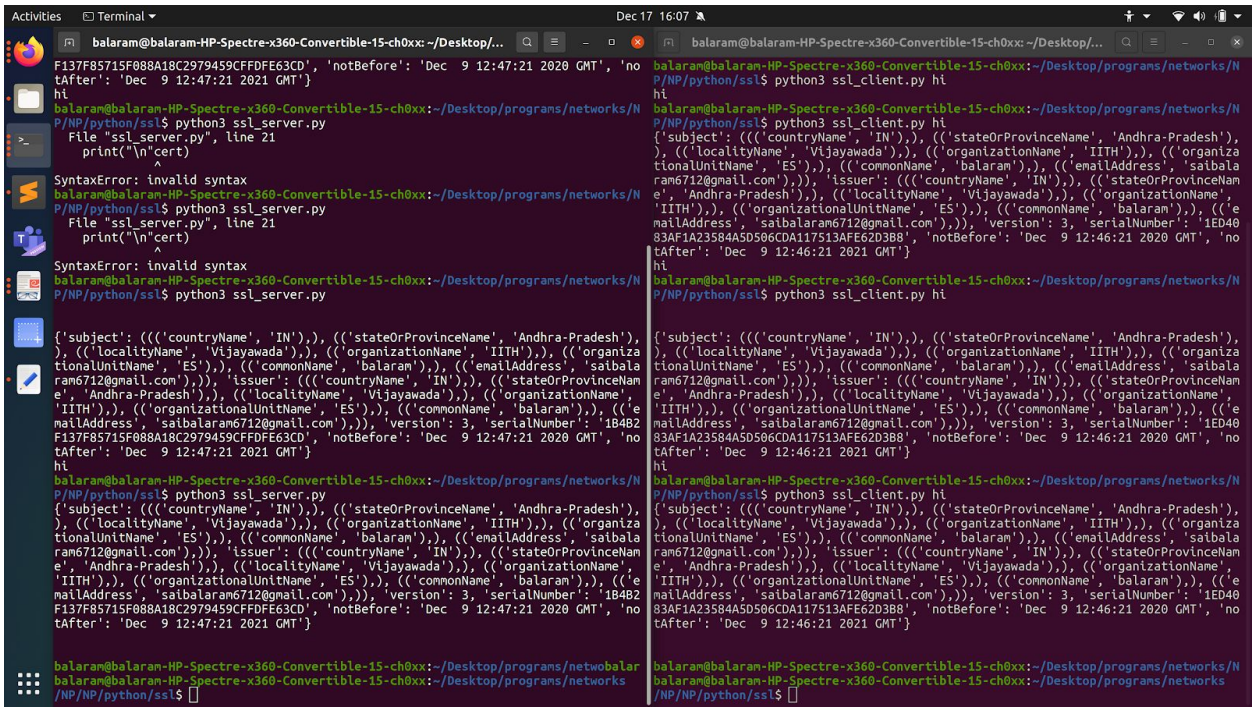
Both the ipv-4 and ipv-6 are printed correspondingly by resolving the hostname using getaddrinfo() from client side

iv) chat room creation using sockets so that multiple people can chat at one time, where a server receives the messages from clients and the message is sent to other clients and also these messages are recorded in the server as well.



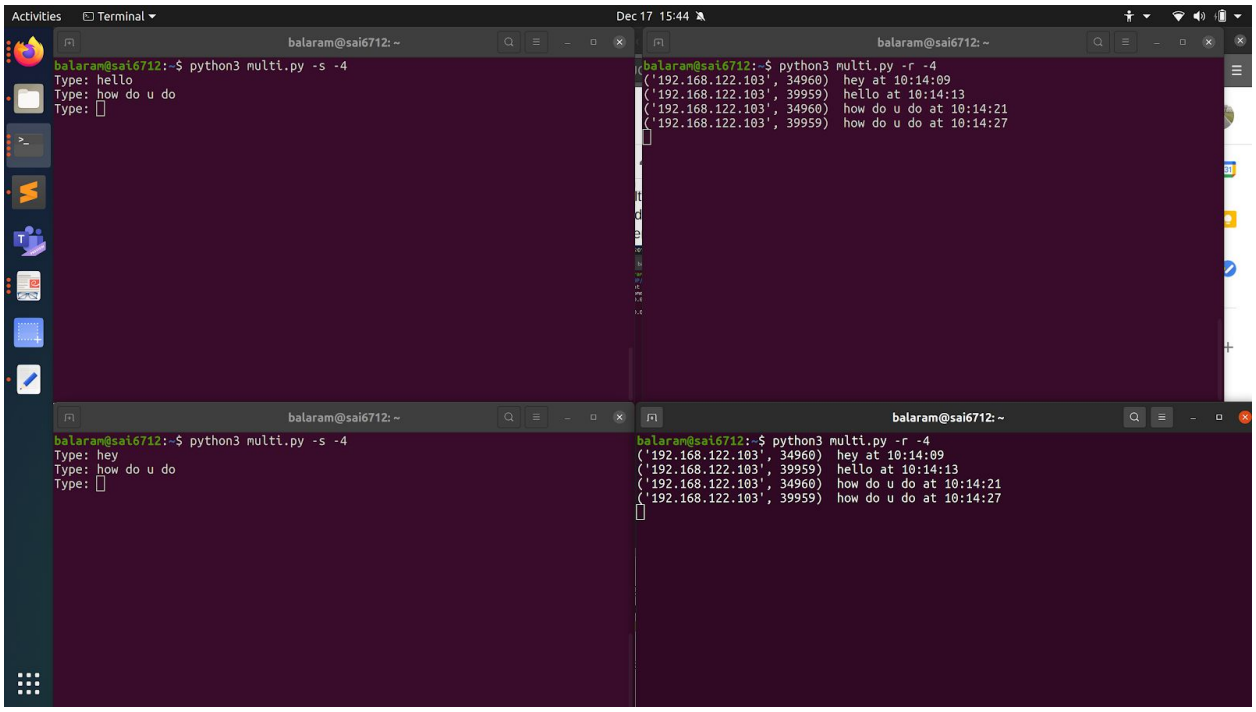
The top left corner in image shows the server receiving messages and printing them and also sending them to all other clients

v) communication from server to client by establishing a handshake connection through ssl by generating certificates and corresponding communication.



Ssl connection and communication through handshake between server and client by sending a sample message echo it

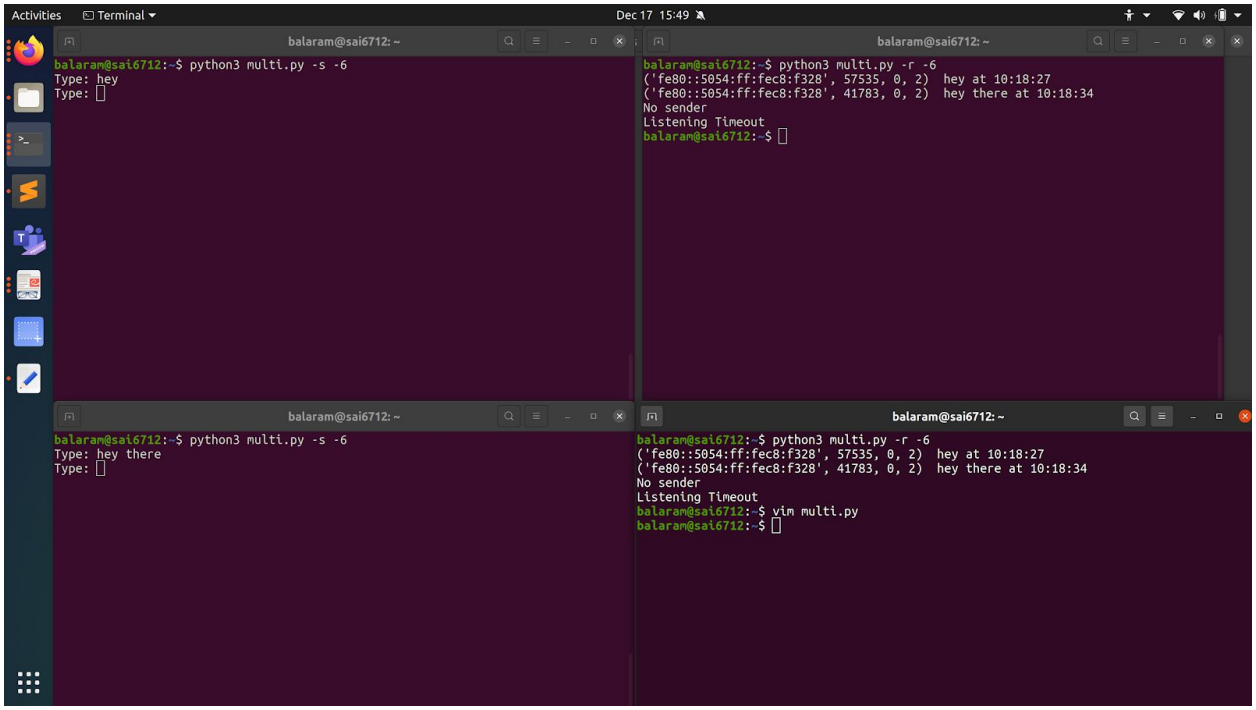
vi) Multicast UDP broadcast where a sender sends the message or any data to a group ip address where some clients or receivers will be receiving on this group ip address to receive the data through that group ip.



The 2 left side terminals are senders to the group sending their message

The 2 right side terminals are receivers to the group which will listen to the group ip and receive correspondingly.

vi)supporting ipv6 for these mentioned functions by mentioning about the usage of ipv6



You can also observe that the multicast supports ipv6 as well and also the time out as there is no response from the senders which can be quite useful for less wastage of time in some cases.

Various files are there in the file with various functions but the main_server and main_client has the features which can be run correspondingly.The ipv6 for all the functions are supported in main_server.py and main_client.py. Broadcast of messages to a group can be quite useful rather than sending the same message multiple times. Ssl connection which allows secure communication is key in terms of privacy
Also some programs also have time out features which end after some time of running if not active things are going on and exit for some of the clients sending messages.Some other features can be ssl file transfer and echo loop and using the connection created in chat mode for secure communication

The readme.md file has the instructions on executing correspondingly the files and also can be observed in the files

Q3: Without explicit mention of ipv4 or ipv6 can be done by running 2 threads in server to resolve the ip address accordingly and getaddrinfo() in client side to create the socket accordingly and implemented an echo server which sends and gets back the text from the terminal side. Present in the files server_gen.py and client_gen.py

Files submitted:

main_server.py, main_client.py, multicast.py, the other functional files

main_server.py and main_client.py contains the functionalities which can be used by passing corresponding arguments

Sources and References:

1) For programming in python using sockets [https://docs.python.org/3/library/socket.html]
2) For ssl [ssl and sockets]
3) For multiple clients [Multithreading python]
4) For some references [https://beej.us/guide/bgnet/html/]
5) getaddrinfo in python [Getaddrinfo function, getaddrinfo another ref]
6) Multicast [https://pymotw.com/2/socket/multicast.html], [Multi cast stack overflow]
7) Time out error [timeout command]
8) Multiple clients [Multi]
9) file transfer [files and sockets]
10) Time https://stackoverflow.com/questions/415511/how-to-get-the-current-time-in-python