

Virtual Chat

# Introduction:

We have implemented a robust Virtual Chat using the concepts of networking paired with socket level programming using Python 2.7.

The main working of the virtual chat is established using a TCP connection between multiple clients (users) and a single server for handling the transactions between them. The users would be able to communicate in real-time, send text messages as well as transfer files throughout the network.

The main concepts used behind establishing a server interface and the clients are given below:

## Server Side:

The server has a fixed IP address, which clients can use to connect to the chat.

1. Socket Level Programming
2. Multi-Threading for handling multiple clients and broadcasting a message
3. String manipulation to implement the client functionalities (as given in section 3.3 of project proposal) i.e. Change name, Quit.

## Client Side:

1. Socket Level Programming
2. Efficient I/O, which handles more than one inputs at a time (messages from server or from the user)
3. Dynamic/Real-Time Client messaging through TCP connection
4. String manipulation to implement the client functionalities (as given in section 3.3 of project proposal) i.e. Blacklist, Unblock, Sleep, Quit.

# Features:

Our Virtual Chat has the following features:

* Each client can:
  + Add his/her own user name (Asked at the start of the session).
  + Change his/her user name (Using the command “\name\<new\_name>”).
  + Mute notifications for a specific time interval (Using the command “\sleep\<time\_to\_sleep>”).
  + Blacklist other users on the network (Using the command “\black\<user\_name>”).
  + Unblock other users on the network (Using the command “\unblock\<user\_name>”).
  + Disconnect without explicitly closing the terminal (Using the command “\quit”).
  + Broadcast files to all other clients.
* The server can:
  + Support multiple clients.
  + Notify the clients on the network that a new client has joined.
  + Handle multiple connection and broadcast requests at the same time through multi-threading.
  + Broadcast text messages sent from a client to all the online clients.
  + Broadcast files sent from a client to all the online clients.