Project Document

BTech/III Year CSE/VI Semester

15CSE312

COMPUTER NETWORK

Chat Server

Group Number – 17

Registration No	Name	Email ID	Contribution
CB.EN.U4CSE18250	A. Sai Tharun	cb.en.u4cse18250@cb.students.amrita.edu	Documentation, Code
CB.EN.U4CSE18258	Sudarshan M.S	cb.en.u4cse18258@cb.students.amrita.edu	Code, Diagrams
CB.EN.U4CSE18259	K. Tanya	cb.en.u4cse18259@cb.students.amrita.edu	Diagrams, Documentation

Amrita School of Engineering, Coimbatore

Department of Computer Science and Engineering

2020 -2021 Even Semester

Table of Contents

To be noted: - At any point of time if you wish to return to this contents page just click on the page number or the footer content.

TITLE OF THE APPLICATION:

Chat Server - Web Application

Humans are social creatures and communication is the basic criterion for social interactions. This makes a proper means of communication a basic human right. To ensure proper and secure communication, we design and build a Chat Server and perform an in-depth analysis of the network related components of it.

The IRC Protocol is an application layer protocol based on the client-server model, and is well suited to running on many machines in a distributed fashion. A typical setup involves a single process (the server) forming a central point for clients (or other servers) to connect to, performing the required message delivery/multiplexing and other functions. Many IT (Information Technology) clients are available for different kinds of computers, so whether you have PC, Macintosh, or UNIX work-section, you will be able to use IRC.

Problem statement:

To understand the working of networks in a chat server, a platform where like-minded people can collaborate, without having the need to authenticate themselves in a place (called chat rooms), and discuss about any topic with privacy. The individual chat servers are connected to each other and thus people can hop from one room to another with ease.

Why Networking is required for the application:

(a.) To allow users to share information and communicate with each other.
(b.) To help like-minded people collaborate with each other.
(c.) To help maintain privacy and establish communication anonymously.
(d.) To have a reliable communication channel, etc.,

How does a user chat with others? / How it works:

A user puts in the web/IP address of the website that serves as the frontend of the chat rooms. In there he is prompted to enter a random name using which he wants to chat. Once he enters the name the system checks if there is already someone currently in any of the rooms with that name and incase a match occurs the user is prompted to choose another name else, he is allowed inside. Once inside the user can either create his own room or join the room (name of the room is displayed) which are already running.
Once he chooses a room he is allowed to enter and he is shown a wall with all the previous conversations that have taken place in the room, from there he can either send a message that is visible to everyone in the room with the name that he has initially chosen or read the conversations that are already going on. Incase the user abuses his privileges and hurts other users; the users can opt to push him out of the room (a rule of vote is considered here).
Thus, the user’s privacy is ensured and he is able to talk to other people.

TECHNICAL CONCEPTS OF THE APPLICATION:

Components of the system: -

1. **Servers: -** These are the backbone of the IRC as it is the only component of the protocol which is able to link all the other components together: it provides a point to which clients may connect to talk to each other (IRC-Client), and a point for other servers to connect to (IRC-Server). The server is also responsible for providing the basic services defined by the IRC protocol.
2. **Clients:** - A Client is anything connecting to a server that is not another server. There are two types of clients, i.e., the User Client and the Service Client
    a. **User Client:** - User clients are generally programs providing a text-based interface that is used to communicate interactively via IRC. This particular type of clients is often referred as "users".
    b. **Service Clients:** - Unlike users, service clients are not intended to be used manually nor for talking. They have a more limited access to the chat functions of the protocol, while optionally having access to more private data from the servers. Services are typically automatons used to provide some kind of service (not necessarily related to IRC itself) to users. An example is a service collecting statistics about the origin of users connected on the IRC network.

Benefits of the proposed system:

(a.) Privacy is ensured
(b.) Entertainment
(c.) Knowledge sharing
(d.) No Tracking
(e.) Can share anything with freedom
(f.) Communicating with like minded people around the world
(g.) Lightweight (does not require heavy hardware configurations)
(h.) Supports almost all the software available.
(i.) Easy to implement.
(j.) Easy to set up and use.
(k.) Accessible from anywhere with just an internet connection

Protocols used:

(a.) Internet Relay chat (IRC)
(b.) Proxy detection
(c.) Transmission control Protocol (TCP) usually on port 6667 but can use ports 6665-6667 and 8000-8002.
(d.) Transport Layer Security (TLS) (Client to Server) on ports 6697, 7000 and 7070.
(e.) Secure Direct Client to Client (SDCC) (Client to Client)
(f.) Internet Protocol (IPv4, IPv6) (to connect with the IRC)
(g.) Channel Hosting and Management

Software/Operating System used:

1. **Operating System:**
   - **a.) Server: -**
     - i.) Windows (ex. mIRC, etc.,)
     - ii.) Linux (ex. Xchat)
     - iii.) Mac OS X
     - iv.) Chrome OS
   - **b.) Client: -**
     - i.) Windows (ex. mIRC, Bersirc, ChatZilla, etc.,)
     - ii.) Linux (ex. Xchat, HexChat, etc.,)
     - iii.) Mac OS X (ex. IRCle, ChatZilla, Fire, X-Chat Aqua, Conversation, etc.,)
     - iv.) Android (ex. AndChat, Holo IRC, IRC Cloud, etc.,)
     - v.) IOS (ex. IglooIRC, Palaver IRC, Colloquy, etc.,)
     - vi.) Chrome OS (ex. CIRC, Byrd, etc.,)

2. **Cloud Services:**

   The traditional IRC does not require any cloud services but a few modern-day IRC's are using a cloud-based approach and are quite successful. Ex. Amazon AWS used by IRCCloud (an IRC application using modern approaches), Azure, etc.,

3. **Programming Languages:**
   - a.) C
   - b.) C++
   - c.) Python
   - d.) Java
   - e.) mIRC scripting language (mSL)
   - f.) Lisp, etc.,

Performance parameters:

Parameter	Meaning	Formula
Bandwidth	Bandwidth is the capacity of a wired or wireless network communications link to transmit the maximum amount of data from one point to another over a computer network or internet connection in a given amount of time	Expressed as bits per second (bps), modern network links have greater capacity, which is typically measured in millions of bits per second (megabits per second, or Mbps) or billions of bits per second (gigabits per second, or Gbps).

Throughput	Throughput measures the percentage of data packets that are successfully being sent; a low throughput means there are a lot of failed or dropped packets that need to be sent again.	
Packet Loss	Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Due to network congestion	Efficiency = 100% * (transferred - retransmitted) / transferred Network Loss = 100 - Efficiency
Transmission time	The time required for transmission of a message depends on the size of the message and the bandwidth of the channel.	Transmission time = Message size / Bandwidth
Propagation Time	Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.	Propagation time = Distance /Propagation speed
Processing Delay	Time taken by the processor to process the data packet is called processing delay.	
Queuing Delay	Time spent by the data packet waiting in the queue before it is taken for execution is called queuing delay.	
Jitter	Jitter is defined as the variation in time delay for the data packets sent over a network. This variable represents an identified disruption in the normal sequencing of data packets. Jitter is related to latency, since the jitter manifests itself in increased or uneven latency between data packets, which can disrupt network performance and lead to packet loss and network congestion. Although some level of jitter is to be expected and can usually be tolerated, quantifying network jitter is an important aspect of comprehensive network	Latency=sum of all delays To measure Jitter, we take the difference between samples, then divide by the number of samples (minus 1).

INDEPTH ANALYSIS OF A SAMPLE APPLICATION (CASE STUDY):

In-depth Analysis of RFC 1459 (An IRC Protocol):

All client-to-server IRC protocols in use today are descended from the protocol implemented in the irc2.4.0 version of the IRC2 server, and documented in RFC 1459.

Since RFC 1459 was published, the new features in the irc2.10 implementation led to the publication of several revised protocol documents (RFC 2810, RFC 2811, RFC 2812 and RFC 2813); however, these protocol changes have not been widely adopted among other implementations.

The basic means of communicating to a group of users in an established IRC session is through a channel.

Channel operators is a client on an IRC channel that has extra privileges so as to manage the channel. An operator typically can Kick a user, ban a user, give another user IRC Channel Operator Status or IRC Channel Voice Status, Change the IRC Channel topic while channel mode +t is set, Change the IRC Channel Mode locks among a few other features.

Users and channels may have **modes** that are represented by single case-sensitive letters. In order to correctly parse incoming mode messages and track channel state the client must know which mode is of which type and for the modes that apply to a user on a channel which symbol goes with which letter. In early implementations of IRC this had to be hard-coded in the client but there is now a de facto standard extension to the protocol called ISUPPORT that sends this information to the client at connect time using numeric 005.

There are also users who maintain elevated rights on their local server, or the entire network; these are called **IRC operators**, sometimes shortened to IRCops or Opers (not to be confused with channel operators). As the implementation of the IRCd varies, so do the privileges of the IRC operator on the given IRCd. RFC 1459 claims that IRC operators are "a necessary evil" to keep a clean state of the network, and as such they need to be able to disconnect and reconnect servers. Additionally, to prevent malicious users or even harmful automated programs from entering IRC, IRC operators are usually allowed to disconnect clients and completely ban IP addresses or complete subnets. Networks that carry services (NickServ et al.) usually allow their IRC operators also to handle basic "ownership" matters. Further privileged rights may include overriding channel bans (being able to join channels they would not be allowed to join, if they were not opered), being able to op themselves on channels where they would not be able without being opered, being auto-opped on channels always and so forth.

A **hostmask** is a unique identifier of an IRC client connected to an IRC server. IRC servers, services, and other clients, including bots, can use it to identify a specific IRC session. The format of a hostmask is nick!user@host. The hostmask looks similar to, but should not be confused with an e-mail address. The nick part is the nickname chosen by the user and may be changed while connected. The user part is the username reported by ident on the client. If ident is not available on the client, the username specified when the client connected is used after being prefixed with a tilde. The host part is the hostname the client is connecting from. If the IP address of the client cannot be resolved to a valid hostname by the server, it is used instead of the hostname.

Some standard modes as specified by RFC 1459 are:

User modes

Letter	Description
i	Invisible—cannot be seen without a common channel or knowing the exact name
s	Receives server notices
w	Receives wallops
o	User is an IRC operator (ircop)

Channel modes

Letter	Symbol	Parameter(s)	Description
o	@	Name of affected user	Channel operator—can change channel modes and kick users out of the channel among other things
s			Secret channel—not shown in channel list or user 'whois' except to users already on the channel
p			Private channel—listed in channel list as "prv" according to RFC 1459
n			Users cannot send messages to the channel externally
m			Channel is moderated (only those who hold channel operator or voice status on the channel can send messages to it)
i			Only users with invites may enter the channel.
t			Only channel operators can change the channel topic.
l		Limit number	Limits number of users able to be on channel (when full, no new users can join)
b		Ban mask (nick!user@host with wildcards allowed)	Bans hostmasks from channel
v	+	Name of affected user	Gives a user voice status on channel (see +m above)
k		New channel key	Sets a channel key such that only users knowing the key can enter

URI scheme:

There are three recognized uniform resource identifier (URI) schemes for Internet Relay Chat: irc, ircs, and irc6. When supported, they allow hyperlinks of various forms, including

irc://<host>[:<port>]/[<channel>[?<channel_keyword>]]
ircs://<host>[:<port>]/[<channel>[?<channel_keyword>]]
irc6://<host>[:<port>]/[<channel>[?<channel_keyword>]]

Some popular IRC networks are:

a.) Freenode – around 90k
b.) IRCnet – around 30k
c.) Efnet – around 18k
d.) Undernet – around 17k
e.) QuakeNet – around 15k
f.) Rizon – around 14k
g.) OFTC – around 13k
h.) DALnet – around 8k

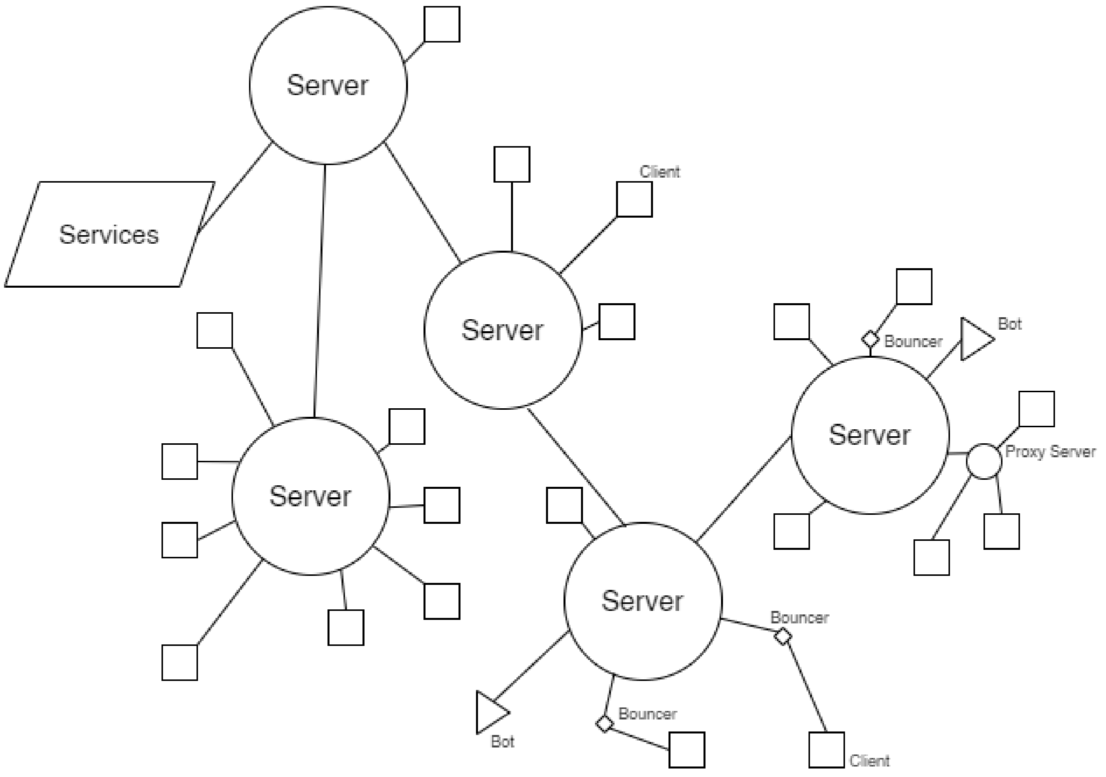
Some Client software:

Chat Type	Software Required
Internet Relay Chat (IRC)	Chat program such as mIRC or Ircle
Web-based Chat	Web browsers like Netscape, Microsoft Edge, Chrome, Firefox, etc.,
AOL (America Online) Chat	AOL access program for America online
Direct Chat Programs	ICQ, AOL Instant Messenger, or another program
Online Conferencing	Conferencing program (CU-SeeMe, Netscape conference, Netmeeting)

Some Communication (Chat) Severs:

Communication servers permit you to give your information to large number of users in environment that is just like Internet newsgroups. The most advanced servers have recently started augmenting text-based medium of conversation with dynamic voice and video support.
There are three major types of communication servers:
1. EFnet servers
2. UnderNet Servers
3. DALnet servers

Basic Architecture diagram:

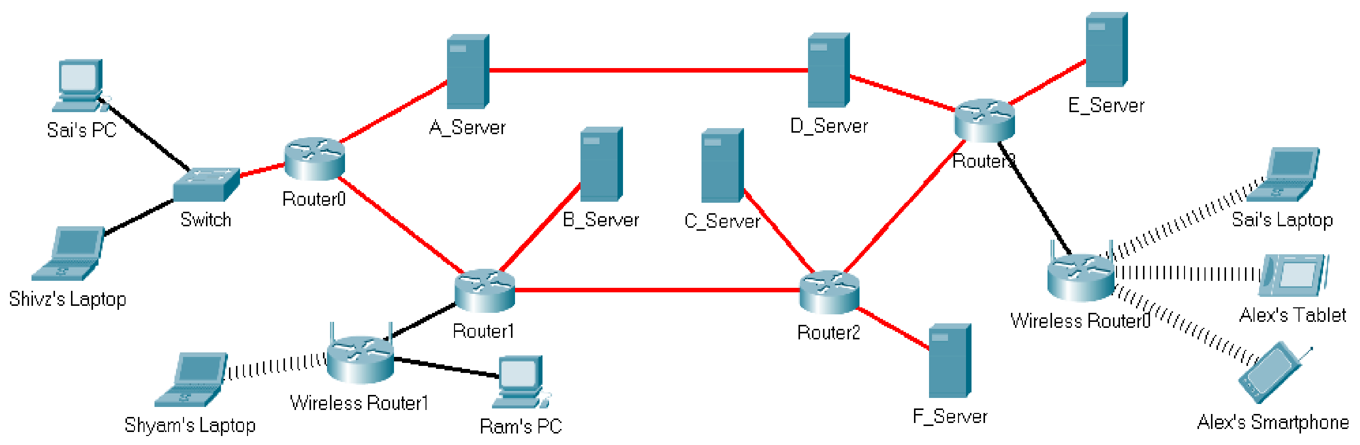


Scheme of an IRC network with servers, normal clients (square), bots (triangle), bouncers (rhombus), additional services and proxy servers.

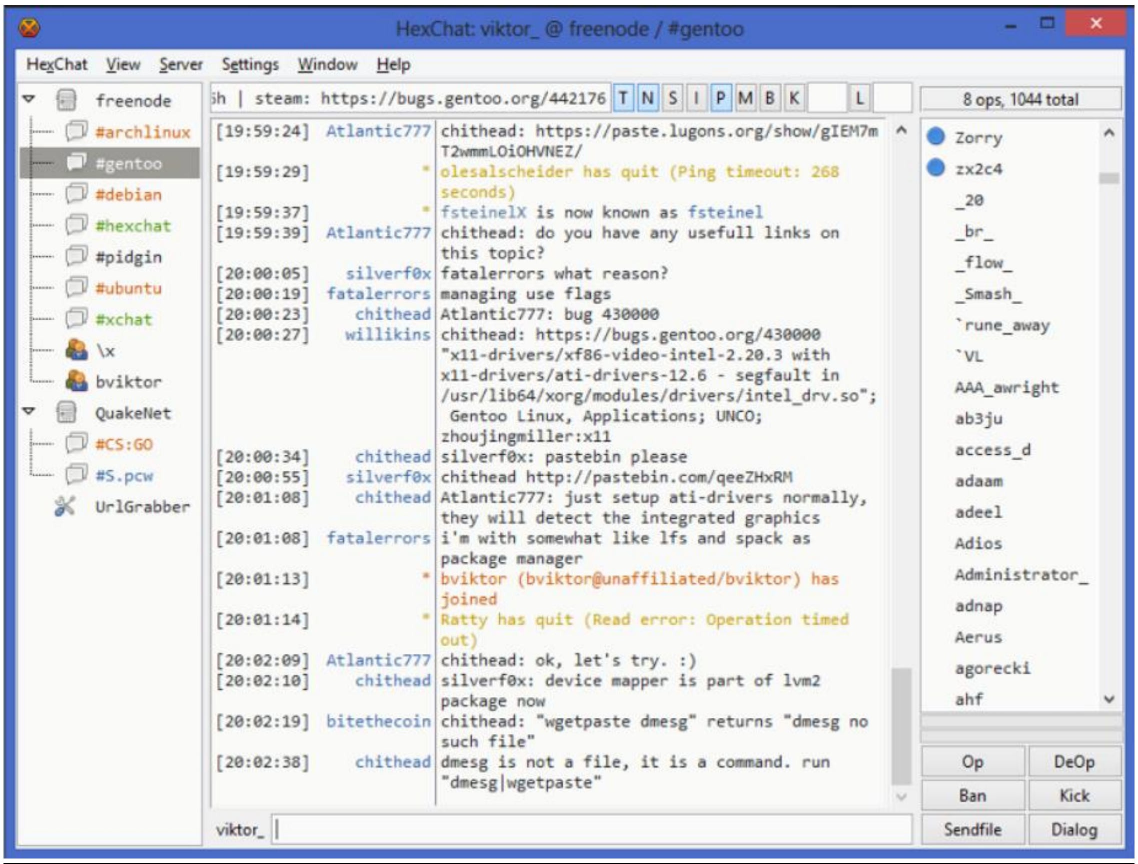
A Small Sample IRC Network

```
1--\
    A        D---4
2--/ \      /
      B----C
     /      \
    3        E

Servers: A, B, C, D, E         Clients: 1, 2, 3, 4
```

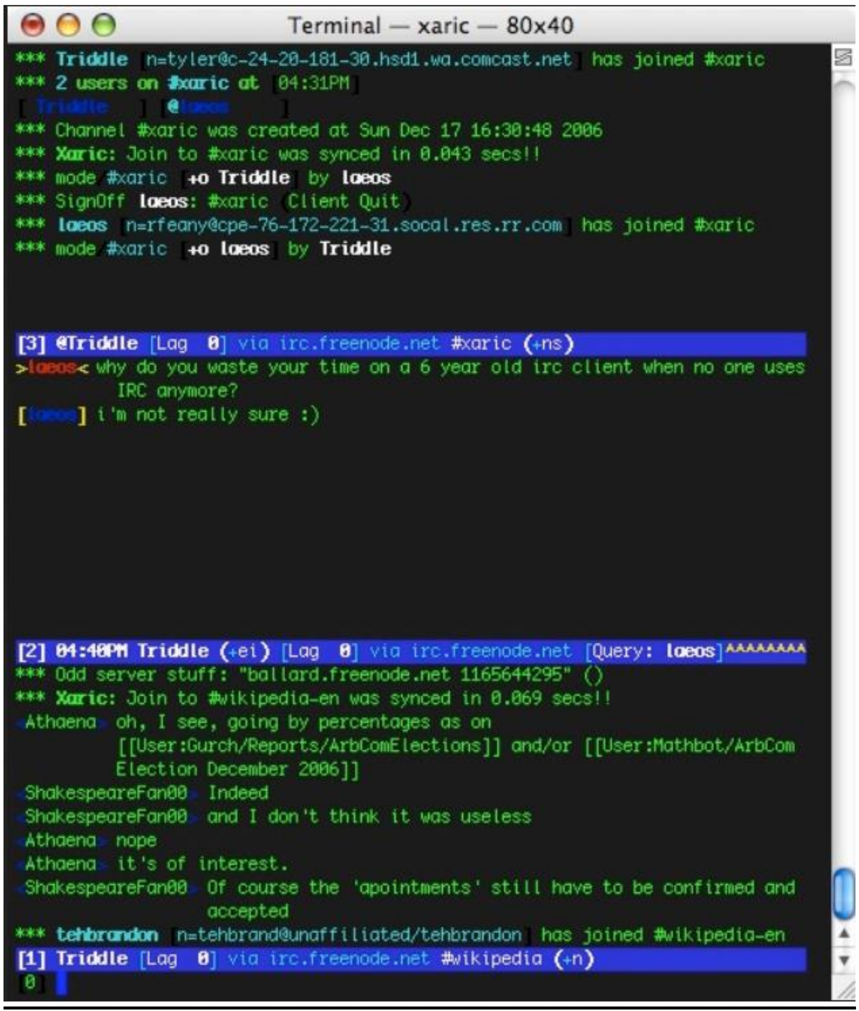
Packet Tracer Diagram



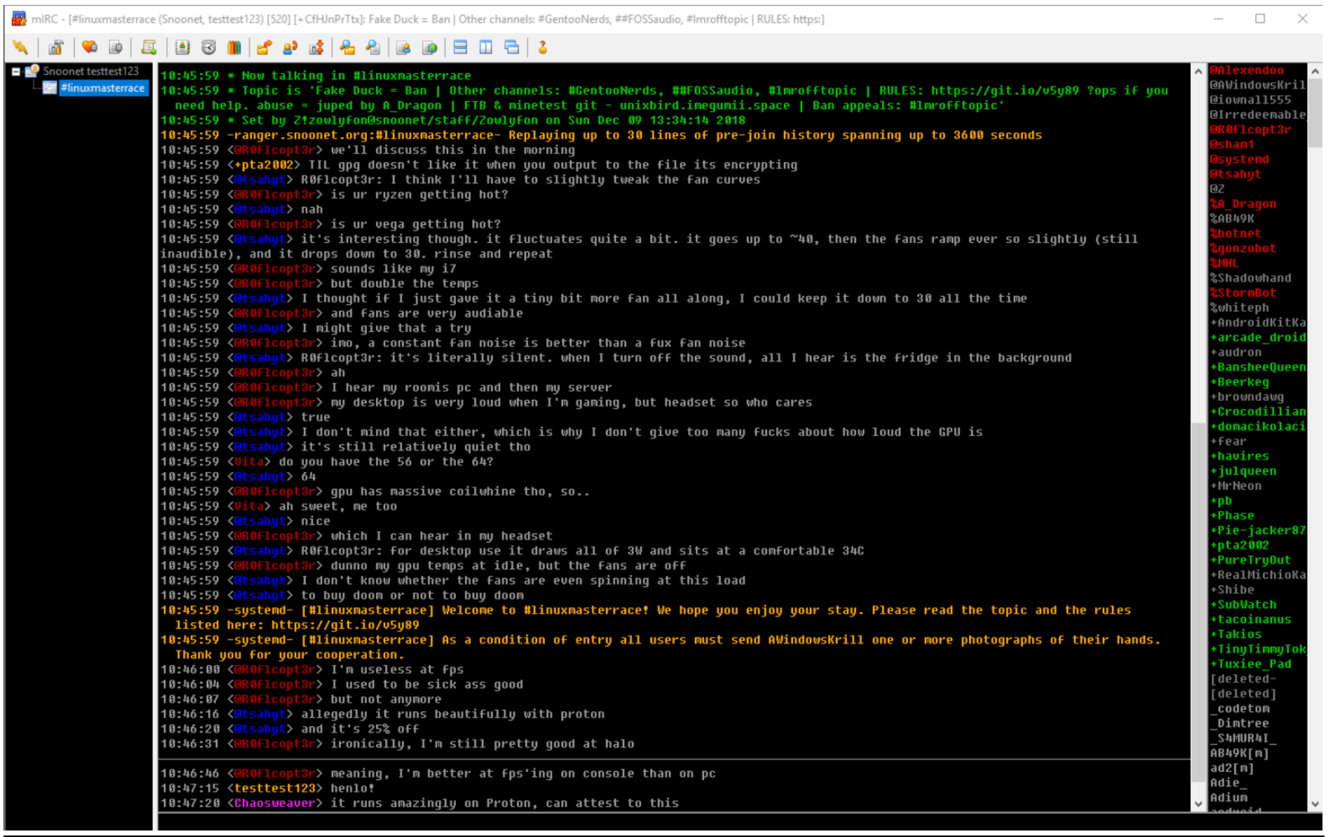
Some pictures of a real applications:



```
ih | steam: https://bugs.gentoo.org/442176
[19:59:24]  Atlantic777 chithead: https://paste.lugons.org/show/gIEM7m
                        T2wmmLOiOHVNEZ/
[19:59:29]            * olesalscheider has quit (Ping timeout: 268
                        seconds)
[19:59:37]            * fsteinelX is now known as fsteinel
[19:59:39]  Atlantic777 chithead: do you have any usefull links on
                        this topic?
[20:00:05]    silverf0x fatalerrors what reason?
[20:00:19]  fatalerrors managing use flags
[20:00:23]     chithead Atlantic777: bug 430000
[20:00:27]    willikins chithead: https://bugs.gentoo.org/430000
                        "x11-drivers/xf86-video-intel-2.20.3 with
                        x11-drivers/ati-drivers-12.6 - segfault in
                        /usr/lib64/xorg/modules/drivers/intel_drv.so";
                         Gentoo Linux, Applications; UNCO;
                        zhoujingmiller:x11
[20:00:34]     chithead silverf0x: pastebin please
[20:00:55]    silverf0x chithead http://pastebin.com/qeeZHxRM
[20:01:08]     chithead Atlantic777: just setup ati-drivers normally,
                        they will detect the integrated graphics
[20:01:08]  fatalerrors i'm with somewhat like lfs and spack as
                        package manager
[20:01:13]            * bviktor (bviktor@unaffiliated/bviktor) has
                        joined
[20:01:14]            * Ratty has quit (Read error: Operation timed
                        out)
[20:02:09]  Atlantic777 chithead: ok, let's try. :)
[20:02:10]     chithead silverf0x: device mapper is part of lvm2
                        package now
[20:02:19]  bitethecoin chithead: "wgetpaste dmesg" returns "dmesg no
                        such file"
[20:02:38]     chithead dmesg is not a file, it is a command. run
                        "dmesg|wgetpaste"
```



```
*** Triddle [n=tyler@c-24-20-181-30.hsd1.wa.comcast.net] has joined #xaric
*** 2 users on #xaric at [04:31PM]
[ Triddle    ] [@laeos      ]
*** Channel #xaric was created at Sun Dec 17 16:30:48 2006
*** Xaric: Join to #xaric was synced in 0.043 secs!!
*** mode/#xaric [+o Triddle] by laeos
*** SignOff laeos: #xaric (Client Quit)
*** laeos [n=rfeany@cpe-76-172-221-31.socal.res.rr.com] has joined #xaric
*** mode/#xaric [+o laeos] by Triddle



[3] @Triddle [Lag  0] via irc.freenode.net #xaric (+ns)
>laeos< why do you waste your time on a 6 year old irc client when no one uses
        IRC anymore?
[laeos] i'm not really sure :)








[2] 04:40PM Triddle (+ei) [Lag  0] via irc.freenode.net [Query: laeos]AAAAAAAA
*** Odd server stuff: "ballard.freenode.net 1165644295" ()
*** Xaric: Join to #wikipedia-en was synced in 0.069 secs!!
<Athaena> oh, I see, going by percentages as on
          [[User:Gurch/Reports/ArbComElections]] and/or [[User:Mathbot/ArbCom
          Election December 2006]]
<ShakespeareFan00> Indeed
<ShakespeareFan00> and I don't think it was useless
<Athaena> nope
<Athaena> it's of interest.
<ShakespeareFan00> Of course the 'apointments' still have to be confirmed and
                   accepted
*** tehbrandon [n=tehbrand@unaffiliated/tehbrandon] has joined #wikipedia-en
[1] Triddle [Lag  0] via irc.freenode.net #wikipedia (+n)
[0]
```



mIRC - [#linuxmasterrace (Snoonet, testtest123) [520] [+CfHJnPrTtx]: Fake Duck = Ban | Other channels: #GentooNerds, ##FOSSaudio, #lmrofftopic | RULES: https:]

Snoonet testtest123
#linuxmasterrace

```
10:45:59 * Now talking in #linuxmasterrace
10:45:59 * Topic is 'Fake Duck = Ban | Other channels: #GentooNerds, ##FOSSaudio, #lmrofftopic | RULES: https://git.io/v5y89 ?ops if you
  need help. abuse = juped by A_Dragon | FTB & minetest git - unixbird.imegumii.space | Ban appeals: #lmrofftopic'
10:45:59 * Set by Z!zowlyfon@snoonet/staff/Zowlyfon on Sun Dec 09 13:34:14 2018
10:45:59 -ranger.snoonet.org:#linuxmasterrace- Replaying up to 30 lines of pre-join history spanning up to 3600 seconds
10:45:59 <@R0flcopt3r> we'll discuss this in the morning
10:45:59 <+pta2002> TIL gpg doesn't like it when you output to the file its encrypting
10:45:59 <@tsahyt> R0flcopt3r: I think I'll have to slightly tweak the fan curves
10:45:59 <@R0flcopt3r> is ur ryzen getting hot?
10:45:59 <@tsahyt> nah
10:45:59 <@R0flcopt3r> is ur vega getting hot?
10:45:59 <@tsahyt> it's interesting though. it fluctuates quite a bit. it goes up to ~40, then the fans ramp ever so slightly (still
inaudible), and it drops down to 30. rinse and repeat
10:45:59 <@R0flcopt3r> sounds like my i7
10:45:59 <@R0flcopt3r> but double the temps
10:45:59 <@tsahyt> I thought if I just gave it a tiny bit more fan all along, I could keep it down to 30 all the time
10:45:59 <@R0flcopt3r> and fans are very audiable
10:45:59 <@tsahyt> I might give that a try
10:45:59 <@R0flcopt3r> imo, a constant fan noise is better than a fux fan noise
10:45:59 <@tsahyt> R0flcopt3r: it's literally silent. when I turn off the sound, all I hear is the fridge in the background
10:45:59 <@R0flcopt3r> ah
10:45:59 <@R0flcopt3r> I hear my roomis pc and then my server
10:45:59 <@R0flcopt3r> my desktop is very loud when I'm gaming, but headset so who cares
10:45:59 <@tsahyt> true
10:45:59 <@tsahyt> I don't mind that either, which is why I don't give too many fucks about how loud the GPU is
10:45:59 <@tsahyt> it's still relatively quiet tho
10:45:59 <Vita> do you have the 56 or the 64?
10:45:59 <@tsahyt> 64
10:45:59 <@R0flcopt3r> gpu has massive coilwhine tho, so..
10:45:59 <Vita> ah sweet, me too
10:45:59 <@tsahyt> nice
10:45:59 <@R0flcopt3r> which I can hear in my headset
10:45:59 <@tsahyt> R0flcopt3r: for desktop use it draws all of 3W and sits at a comfortable 34C
10:45:59 <@R0flcopt3r> dunno my gpu temps at idle, but the fans are off
10:45:59 <@tsahyt> I don't know whether the fans are even spinning at this load
10:45:59 <@tsahyt> to buy doom or not to buy doom
10:45:59 -systemd- [#linuxmasterrace] Welcome to #linuxmasterrace! We hope you enjoy your stay. Please read the topic and the rules
  listed here: https://git.io/v5y89
10:45:59 -systemd- [#linuxmasterrace] As a condition of entry all users must send AWindowsKrill one or more photographs of their hands.
  Thank you for your cooperation.
10:46:00 <@R0flcopt3r> I'm useless at fps
10:46:04 <@R0flcopt3r> I used to be sick ass good
10:46:07 <@R0flcopt3r> but not anymore
10:46:16 <@tsahyt> allegedly it runs beautifully with proton
10:46:20 <@tsahyt> and it's 25% off
10:46:31 <@R0flcopt3r> ironically, I'm still pretty good at halo

10:46:46 <@R0flcopt3r> meaning, I'm better at fps'ing on console than on pc
10:47:15 <testtest123> henlo!
10:47:20 <Chaosweaver> it runs amazingly on Proton, can attest to this
```

```
@Alexendoo
@AWindowsKril
@iownall555
@Irredeemable
@R0flcopt3r
@sham1
@systemd
@tsahyt
@Z
%A_Dragon
%AB49K
%botnet
%gonzobot
%MHL
%Shadowhand
%StormBot
%whiteph
+AndroidKitKa
+arcade_droid
+audron
+BansheeQueen
+Beerkeg
+browndawg
+Crocodillian
+domacikolaci
+fear
+havires
+julqueen
+MrNeon
+pb
+Phase
+Pie-jacker87
+pta2002
+PureTryOut
+RealMichioKa
+Shibe
+SubWatch
+tacoinanus
+Takios
+TinyTimmyTok
+Tuxiee_Pad
[deleted-
[deleted]
_codetom
_Dimtree
_S4MUR4I_
AB49K[m]
ad2[m]
Adie_
Adium
```

SAMPLE IMPLEMENTATION:

Code: -

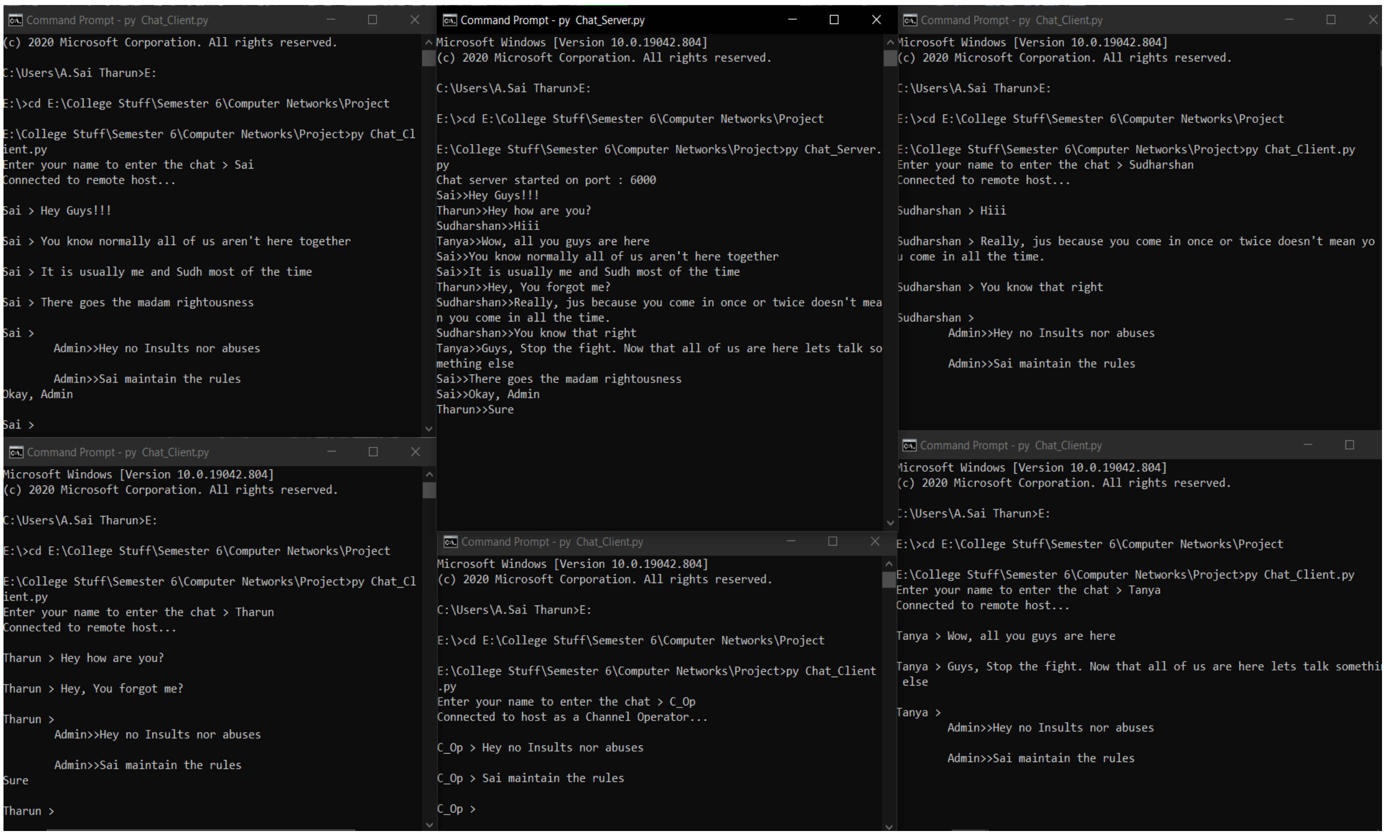
Chat_Server.py

```
import socket, threading

def accept_client():
    while True:
        cli_sock, cli_add = ser_sock.accept()
        CONNECTION_LIST.append(cli_sock)
        thread_client = threading.Thread(target = msgrecieve_usr, args=[cli_sock])
        thread_client.start()

def msgrecieve_usr(cli_sock):                                   # Recieving the msg
    while True:
        try:
            data = cli_sock.recv(1024)
            if data:
                name = data.decode()
                uname = name.split('>>')[0]
                if uname == 'C_Op':
                    msg = 'Admin'+'>>'+name.split('>>')[1]
                    b_usrs(cli_sock, msg)
                else:
                    m_usr(cli_sock, data)
        except Exception as x:
            print(x)
            break

def m_usr(cs_sock, msg):                                        # Printing the msg sent by the user
    for client in CONNECTION_LIST:
        if client == cs_sock:
            print(msg.decode())

def b_usrs(cs_sock, msg):                                       # Broadcasting the msg sent by the admin
    for client in CONNECTION_LIST:
        if client != cs_sock:
            client.send(msg.encode())


if __name__ == "__main__":
    CONNECTION_LIST = []
    ser_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)     # Creating a socket

    HOST = 'localhost'                                          # Binding a socket
    PORT = 6000
    ser_sock.bind((HOST, PORT))

    ser_sock.listen(1)
    print('Chat server started on port : ' + str(PORT))

    thread_ac = threading.Thread(target = accept_client)
    thread_ac.start()
```

Chat_Client.py

```
import socket, threading
def send(uname):                                                   # Sending the msg to the server
    while True:
        msg = input(f'\n{uname} > ')
        data = uname + '>>' + msg
        cli_sock.send(data.encode())

def receive():                                                     # Recieving the broadcast msg from the server
    while True:
        data = cli_sock.recv(1024)
        print('\n\t'+ str(data.decode()))

if __name__ == "__main__":
    cli_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)   # Creating a socket

    HOST = 'localhost'                                             # Connecting to the socket
    PORT = 6000
    cli_sock.connect((HOST, PORT))

    uname = input('Enter your name to enter the chat > ')
    if uname == 'C_Op':
        print('Connected to host as a Channel Operator...')
    else:
        print('Connected to remote host...')

    thread_send = threading.Thread(target = send,args=[uname])
    thread_send.start()
    thread_receive = threading.Thread(target = receive)
    thread_receive.start()
```

Output: -



```
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Cl
ient.py
Enter your name to enter the chat > Sai
Connected to remote host...

Sai > Hey Guys!!!

Sai > You know normally all of us aren't here together

Sai > It is usually me and Sudh most of the time

Sai > There goes the madam rightousness

Sai >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
Okay, Admin

Sai >
```

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Server.
py
Chat server started on port : 6000
Sai>>Hey Guys!!!
Tharun>>Hey how are you?
Sudharshan>>Hiii
Tanya>>Wow, all you guys are here
Sai>>You know normally all of us aren't here together
Sai>>It is usually me and Sudh most of the time
Tharun>>Hey, You forgot me?
Sudharshan>>Really, jus because you come in once or twice doesn't mea
n you come in all the time.
Sudharshan>>You know that right
Tanya>>Guys, Stop the fight. Now that all of us are here lets talk so
mething else
Sai>>There goes the madam rightousness
Sai>>Okay, Admin
Tharun>>Sure
```

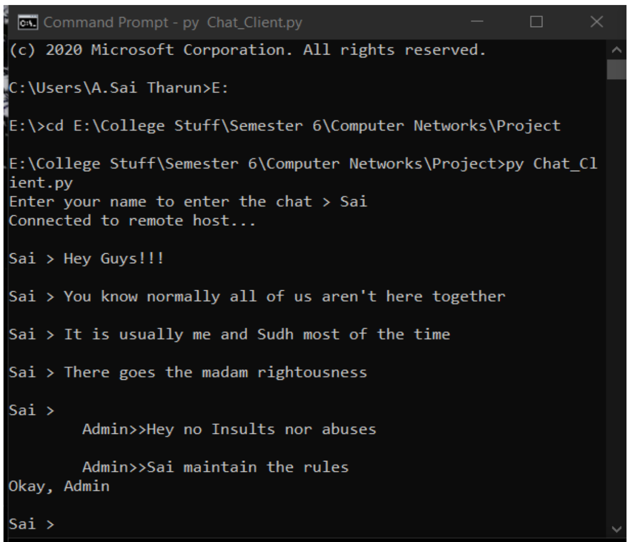
```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Client.py
Enter your name to enter the chat > Sudharshan
Connected to remote host...

Sudharshan > Hiii

Sudharshan > Really, jus because you come in once or twice doesn't mean yo
u come in all the time.

Sudharshan > You know that right

Sudharshan >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
```

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Cl
ient.py
Enter your name to enter the chat > Tharun
Connected to remote host...

Tharun > Hey how are you?

Tharun > Hey, You forgot me?

Tharun >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
Sure

Tharun >
```

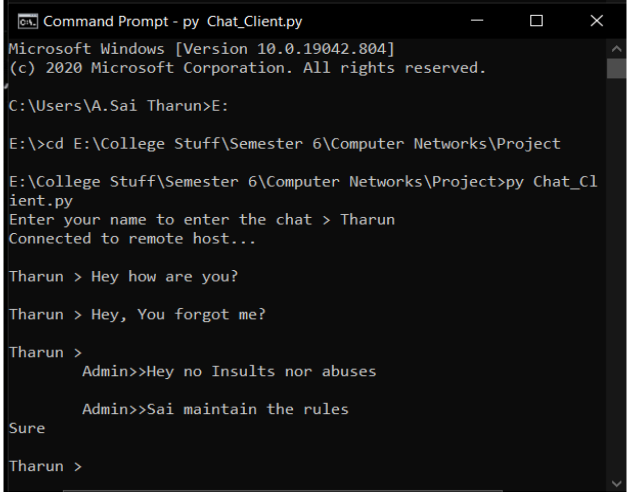
```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Client
.py
Enter your name to enter the chat > C_Op
Connected to host as a Channel Operator...

C_Op > Hey no Insults nor abuses

C_Op > Sai maintain the rules

C_Op >
```

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Client.py
Enter your name to enter the chat > Tanya
Connected to remote host...

Tanya > Wow, all you guys are here

Tanya > Guys, Stop the fight. Now that all of us are here lets talk somethi
 else

Tanya >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
```

(All the Screen together - Center-top is the Server and the center-bottom is the Channel Operator/Admin, rest are users)

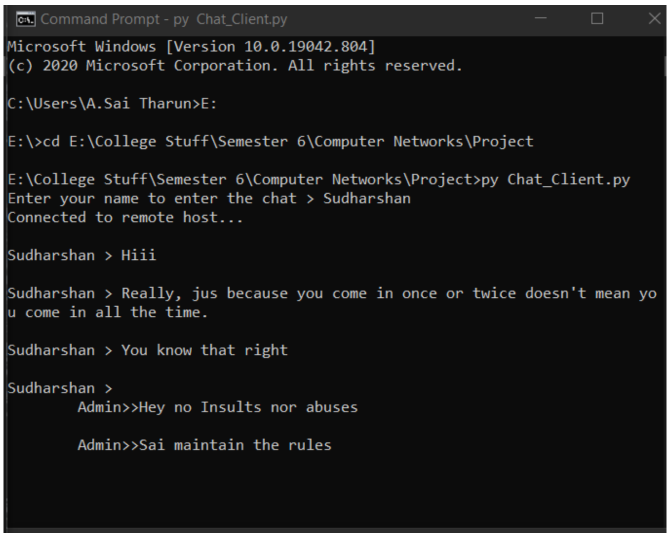


```
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Cl
ient.py
Enter your name to enter the chat > Sai
Connected to remote host...

Sai > Hey Guys!!!

Sai > You know normally all of us aren't here together

Sai > It is usually me and Sudh most of the time

Sai > There goes the madam rightousness

Sai >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
Okay, Admin

Sai >
```

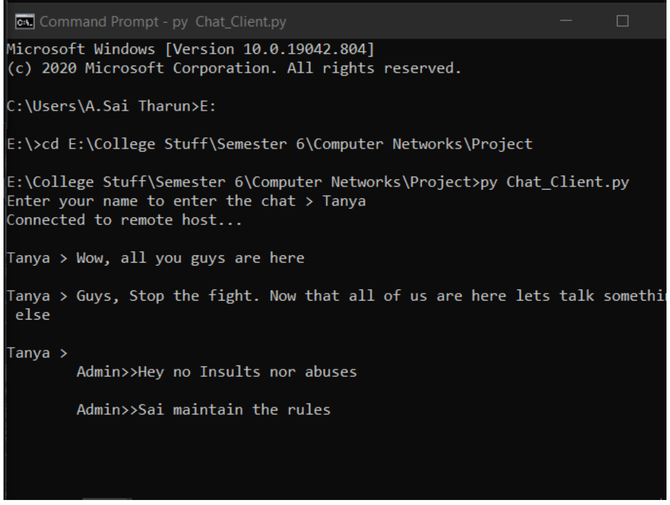


```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Cl
ient.py
Enter your name to enter the chat > Tharun
Connected to remote host...

Tharun > Hey how are you?

Tharun > Hey, You forgot me?

Tharun >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
Sure

Tharun >
```

(1st User – Sai and the 2nd User – Tharun Window)

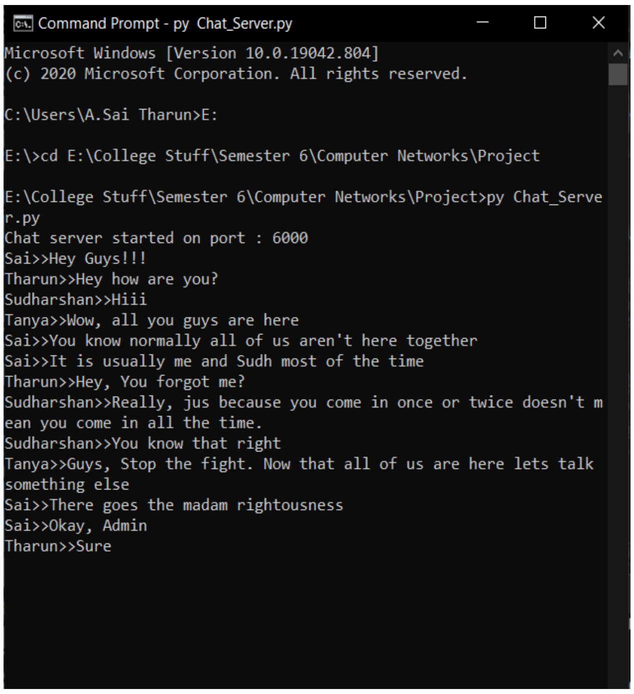


```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Client.py
Enter your name to enter the chat > Sudharshan
Connected to remote host...

Sudharshan > Hiii

Sudharshan > Really, jus because you come in once or twice doesn't mean yo
u come in all the time.

Sudharshan > You know that right

Sudharshan >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
```



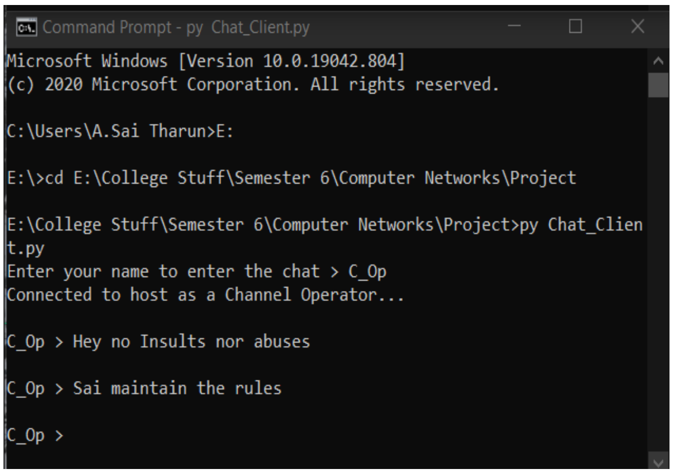
```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Client.py
Enter your name to enter the chat > Tanya
Connected to remote host...

Tanya > Wow, all you guys are here

Tanya > Guys, Stop the fight. Now that all of us are here lets talk somethi
 else

Tanya >
        Admin>>Hey no Insults nor abuses

        Admin>>Sai maintain the rules
```

(3rd User – Sudharshan and the 4th User – Tanya Window)



```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Serve
r.py
Chat server started on port : 6000
Sai>>Hey Guys!!!
Tharun>>Hey how are you?
Sudharshan>>Hiii
Tanya>>Wow, all you guys are here
Sai>>You know normally all of us aren't here together
Sai>>It is usually me and Sudh most of the time
Tharun>>Hey, You forgot me?
Sudharshan>>Really, jus because you come in once or twice doesn't m
ean you come in all the time.
Sudharshan>>You know that right
Tanya>>Guys, Stop the fight. Now that all of us are here lets talk
something else
Sai>>There goes the madam rightousness
Sai>>Okay, Admin
Tharun>>Sure
```

(Chat Server Window)



```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\A.Sai Tharun>E:

E:\>cd E:\College Stuff\Semester 6\Computer Networks\Project

E:\College Stuff\Semester 6\Computer Networks\Project>py Chat_Clien
t.py
Enter your name to enter the chat > C_Op
Connected to host as a Channel Operator...

C_Op > Hey no Insults nor abuses

C_Op > Sai maintain the rules

C_Op >
```

(Channel Operator/Admin Window)

ANALYTICAL QUESTIONS:

1. What are the likely problems users can face?

A. Users can face issues while the initial set up and in accessing the channels for the first time. The users would also have to go through the guidelines that are set up for the channel operators to prevent themselves from being kicked out of the channel. The users might also face the trouble of sending messages due to congestion in the server, which happens when many people access the server at the same time (more than the server can handle)

2. How does the application tackle the above-mentioned problems?

A. For the problem with network congestion, we can reduce it through network segmentation and content delivery network, we can have a upper limit over the number of people who can connect to the server at a time in a first come first serve basis or whatever method we see fit and let the remaining connections wait until anyone already in the channel leaves and any other issues can be tackled by developing a system that is logically implemented well to take care of redundancy and error correction and reduction.

3. Is the application flexible according to the customer?

A. The application can be made flexible according to the user base the application is being developed for. We can have it as a terminal operated application, or a web application, or we can also implement it as a GUI based standalone application. We can also build it with custom level of privacy based on the user base the application is being developed for.

4. What is the architecture used to implement IRC?

A. Servers are connected in spanning tree fashion architecture. In this, each server is connected to several others, which may be another server/set of servers or a number of clients. Server-Server connection is used for server-server communication (relying messages), to increase the capacity and length of the network (to have a greater ease and to adjust traffic flow properly) while Server-Client is used for communication purposes (the main objective for building the application). When you connect to server, first you have to choose specific channel to join and choose user name to identify yourself when you at chat. Your message is sent from client software on your PC to IRC server to which you are connected. Then message is sent from one server to other servers where all users on this channel are logged on.

5. Is there a limit on the number of channels a user can join/create?

A. There is no restriction on the number of channels a user can join or create, but then the server hosting the channels should have the required capacity to accommodate them i.e., the number of channels a user can create and host is limited by the hardware capacity of the server system hosting the channels.

6. Is it costly to implement and run a chat server?

A. Implementing a chat server is relatively cheap and can be done by anyone with basic knowledge of networking concepts. Cost depends on the number of clients you wish to be able to accommodate in your server and the purpose of creating it. So, the cost can be anywhere between a few dollars to a few hundred dollars. The majority cost incurred is only during creation of the chat server, once establish the operational costs are very minimal.

7. Does it have privacy features?

A. The application is built for privacy. Basic privacy features are already there but many other privacy centric features can be added based on the user base the application is being developed for.

8. Can I communicate with another client one on one?

A. That is not the main feature of the application but then we can add such a functionality too. It is possible to implement such a functionality wherein the server does not display the message sent by the client on it's wall but instead relays it back to the client it was intended to reach.

9. Is the application secure?

A. Yes, the application is secure with basic security features in place. Incase more security features are required they can always be implemented based on the requirement.

10. Is the application scalable?

A. Yes, the application can be scaled according to the future requirements. We can add extra hardware to the server to scale it or we can add another server to support the first one (can expand to any size this way) by adequately splitting the traffic between the servers and a few other methods are also possible to help us scale the application.

11. Can I send anything other than text like pictures, audio, video or other files?

A. IRC was built for text (chat) based communication and so has protocols only meant for that but it can always be expanded to accommodate files of various formats others than text. To accomplish that we would have to add in some more protocols and do some additions in the server code, possibly we would have to provide addition methods like compressing, resampling, etc., to prevent huge files sent by client from clogging the bandwidth and the memory of the server.

12. Why such a solution has been proposed to tackle the problems?

A. This solution has been proposed to facilitate easy communication between people with minimal training and infrastructure required from the user end. This solution also provides privacy to the user and promotes freedom of speech along with various other benefits. It's easy to implement, scale up and so can accommodate a large number of users thereby facilitating easy collaboration between people with common ideas from anywhere across the globe. Since the messages are stored in the server, a user can access the messages from anywhere irrespective of the location, and device being used even with a low internet bandwidth.

SOME CHALLENGES TO BE FACED:

Even though IRC is so effective and provides a host of features that ensure privacy, freedom of speech, sharing ideas, etc., these very features are misused for nefarious purposes by some people which harms the other users who want to use the application for good. Some of the prominent challenges faced are

Attacks

Since IRC connections are usually unencrypted and span long time periods, they are usually targeted by DoS/DDoS attackers and hackers. To overcome this IRC servers, use SSL/TLS connections, while this helps stop the use of packet sniffer programs to obtain the passwords of IRC users, has little use beyond this scope due to the public nature of IRC channels. SSL connections require both client and server support (that may require the user to install SSL binaries and IRC client specific patches or modules on their computers). IRC in the past served as a laboratory for many kinds of Internet attacks, such as using fake ICMP unreachable messages to break TCP-based IRC connections (nuking) to annoy users or facilitate takeovers.

Abuse Prevention

Since the application does not take the credentials of the user (to provide privacy) the user is free to talk anything he wishes to, and sometimes some users tend to misuse the freedom to abuse other users. It is the job of the server admin or channel operator to ban such users (which calls for the need of some kind of a security operator, in some cases such personnel might not be available). There have been incidents where users have been bullied by the other users.

Illegal/Anti-National Activities

Since IRC is secure and does not usually track people, it has been used for promoting Illegal and Anti-national activities with the officials not being able to track the people involved. Once the server is closed all the conversation is lost and is usually not recoverable. There have been instances where people have sold all illegal stuff through IRC, there have also been instances where people have spread Anti-National sentiments and used IRC to recruit people into illegal anti-government organizations in the past.