Indian Institute of Technology Kanpur

Department of Computer Science and Engineering

Computer Networks : CS425A

Multi User Chat Application

Prepared By:

GROUP NUMBER:-	9
Avinash Choudhary	14158
Sagar Chand	14579
Shivam Yadav	14655

Supervisor:

Professor Dheeraj Sanghi

1 Problem Statement

Developing a comprehensive client-server messaging application which has-
• basic security features of a chat room
• support for broadcast and point to point messages
• concepts of asynchronous messaging

2 Objective

We wanted to create a chat app which can be used for communication among various clients having authentication, broadcast message and private message features. Other features like multiple logins , blocking of a user for 60 seconds for 3 consecutive failed attempts, checking who else is currently online and who else was online within last hour have also been implemented. Blocking and unblocking of a user by another have been implemented. Concept of asynchronous messages have been used in private messages.

3 Assumptions

Following assumptions have been made-
• New user registrations are done explicitly from "users.txt" file
• Clients know the valid command line format to connect with server

4 Architecture

Server is turned on a particular IP and port number. Clients connect to this IP and port.
• Security - No user can log in twice. 3 failed login attempts, blocks that user for 60 sec. He/she cannot connect to server during this time.
Once user has connected to server, he/she has list features which he/she can avail of. Following list presents Messaging features-
• Broadcast - type "broadcast <message>" and message goes to all online users
• Private message - type "message <user><message>" and message goes to the particular user. Concept of asynchronous messaging is used here. If user is online, message goes immediately to him/her. If not, then message is stored in a file and sent to the user whenever he/she come online next(and message is deleted from that file)
• Block and Unblock - type "block <user>" to block a user. type "unblock <user>" to block a user. – The users can block and unblock other user(s) Based on this the private messages will be delivered
Other features have also been implemented-
• Logout-User also has option of logging out. By typing "logout" he would be logged out.
• whoelse - By typing "whoelse", user can know all other users who are currently online with him
• wholasthr: By typing "wholasthr", user can know all other users who were connected within last hour.
All these details are shown in flow-diagram (figure - 1. Module Diagram)

Flow diagram for chatpot Project

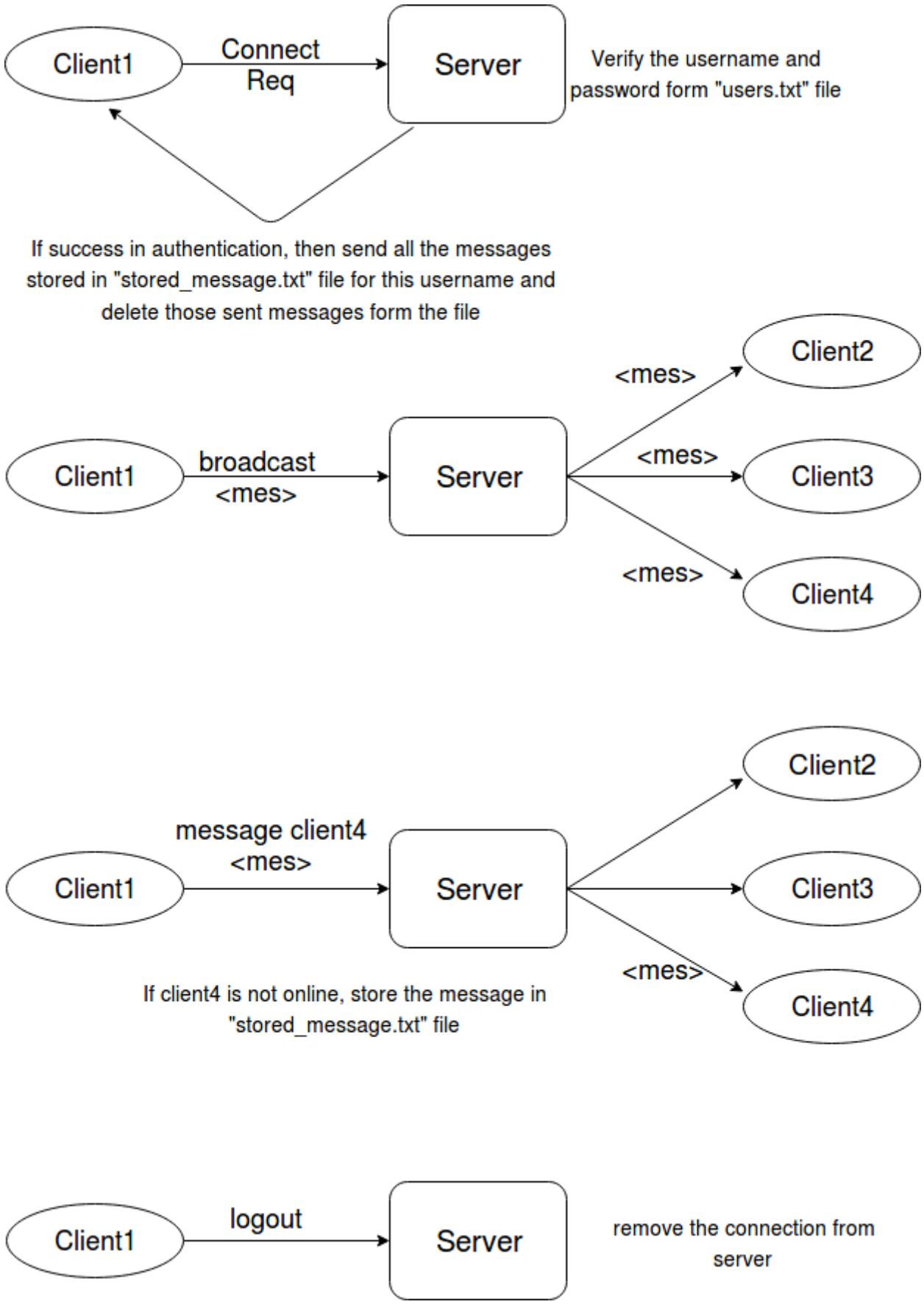

Figure 1: Module Diagram

5 Implementation environment

We have implemented code in Python(python 2.7.14). socket, select, thread, sys and datetime libraries have been used. Code has been written and tested in linux(ubuntu) machines with one machine having the private IITK IP address acting as a server and other machines connecting to this server having private address in this linux machine's network. Thus with (at least)3 different IPs, we were able to test our code. All machines had 4GB RAM and intel i5 processor.

6 Summary

We were able to complete all the tasks we hoped for and even added extra security feature of blocking the user if there is 3 failed login attempts, no matter what IP he uses to login. This way, someone trying to guess his/her password, by popping from multiple IPs is solved. Multiple users were able to chat together and were able to fully utilize all the features. Blocking and unblocking of a user to another user has also been successfully implemented.

References

[1] Help from man pages, stackoverflow and various python documentations are taken for the completion of this project.

[2] Geeksforgeeks is used for basic syntax for connect, bind, listen, etc in python.