

Project 01 - Chat Client and Server

Due: Mar 11 11:59pm

In this project you will create two different chat programs, one will operate as in a server-client model using TCP and one will operate as a P2P system over UDP.

In both cases you will have to develop your own protocol and document them in enough detail so that someone else could implement them. The protocol you make must be general enough so that other vendors can make their own clients with fancy UIs. I recommend documenting the protocols before you get started on the actual programs. You can choose to document this in any format you believe conveys the concepts well enough to other students in the class. This may be a text file, markdown, or even a PDF with diagrams.

All programs must work as command-line programs like the homeworks we have done. You can use the argparse library like they do.

Make sure to test extensively. As part of the assignment, you must setup a Raspberry Pi to run the servers so that they are always running and you can test with at least 2 clients (preferably you will get some friends to help test them out with you). Anyone in Sys Admin should be able to make it so this server auto-starts with the Raspberry Pi, but this won't be part of the assignment.

Additionally you must have good code style, including:

- Well divided into functions
- Each function has documentation
- Well commented
- Good variable and function names

Server-Client Chat System using TCP

In this model you will have a server program that runs a single machine, listens for and accepts incoming connections from any number of simultaneous clients. When a client connects it must provide a unique username (i.e. one that isn't in use by any other client at that time). New clients receive the 10 most recent messages (if there are less than that since the server restarted, then as many there have been). Each of those messages includes the username of the person who sent it and the time at which it was received.

After a new client is setup it can send messages. When the server receives messages for a client it sends that message out to all connected clients (including the client that sent the message). When the server sends this message out, it also includes the username and time. When printing out messages, the username and time are also shown.

P2P Chat System using UDP

In this model you will write a single program (instead of split server/client like for TCP). This setup must have a similar behavior to the TCP model. Upon the first connection, a new peer must announce its chosen username and if any current peer thinks it shouldn't be allowed, the new peer is rejected. Afterwards, the 10 most recent messages (up to however many the oldest peer knows). Each message includes the username and time it was initially sent.

When a peer has been setup it can then simply broadcast new messages including its username and the current time.

Rubric

TCP Protocol Documentation	15
TCP Client	10
TCP Server	15
UDP Protocol Documentation	15
UDP Program	30
Code Style	15