P2P Chatroom Using Sockets in C++

Jonathan Hovick & Ethan Lynch

December 5, 2019

**Overview**

For our final project we decided to do a chatroom application using a peer to peer structure. Because the application is meant to be peer to peer, we had to find a method to make the room server not run on a dedicated computer. To do this, we had one of the clients act as server hosting the messages for the room. When a room is created, the client that requests the creation logs its IP information on an entry server and creates a thread acting as a server for the room. Then each subsequent room requests the information by room name from the static server and requests to connect to the chat room.

**Creating a Room**

When a client wants to create a room, a creation request is sent to the server with the name of the room. If the room name is taken, the server will tell the client and a different room name will have to be chosen. If the room name is not taken, the server will capture the client’s IP address information and create the room; a confirmation message will be sent to the client and the room will be created with that client as its host.

**Joining a Room**

When a client wants to join a room, a get request is sent to the server with the name of the room. If the room exists, the server will send the host IP address information and the client will capture it. If the room does not exist, the server will send a message to the client and the client will have to try a different room name. After the client has entered an acceptable room name, they will send a username to the host of the room. The host will then create a map for that username with the associated client IP address information.

**Sending Messages**

Once all clients are in the chat room, the static server isn’t used, and all communications take place between the clients. When any client (including the host) wishes to send a message, a string containing a marker character, the username, and the message is sent to the host who then distributes the message to the other clients by iterating through a hash map of their IP information. The messages sent between peers are sent using a UDP connection to minimize the number of threads and sockets needed for communication.

**Exiting a Room**

If a client wants to exit a room, they will exit by typing “/exit” in the chat room input. This will cause the client to leave the room and close the application.

**The Future**

Further plans to make this project better include using symmetric key encryption using the password to increase the security of the room, not by denying access but by allowing all messages to be scrambled, creating a GUI for easier usage, host migration so the host peer can leave without having to create a new room, and adding file sharing capabilities.