Ömer Mesud TOKER – 21302479 – Section 1

Salih CAN – 21301963 – Section 1

CS 421 Computer Networks – Programming Assignment 1

Instant Message App

1) <u>In the listen side:</u>

Firstly, we create TCP and UDP sockets. TCP is for sending the HTTP POST request. UDP is for the receiving the messages from other clients. Then, we get the port number for the UDP socket from the operating system. Then, we prepare the packet for the HTTP POST request which has source IP address, data and data length. After preparing the packet, we send it to the server. After that, we receive the message from the server for status code. If the status code is OK (200), we enter an infinite loop and wait for the UDP messages from other clients. Here, to determine the username of the sender, we look at the beginning of the data until the '~' character because we add the username in front of the data in the sender side to determine the name of the sender.

<u>In the send side:</u>

In sender mode, we first analyze the input command that user enters. We break that command into command type, usernames and message. Then, we enter the read-evaluate-print loop and determine the action we take according to command type. In each command, except exit, we need to update the local registry for userlist. To do that, we wrote a method for HTTP GET request and call that method in each command type. In list command, we simply list the online users with HTTP GET request which contains only source IP address and path. Here, we also check for the status code of the GET request then, we store the username, IP address and port numbers locally. In the unicast command, we update the registry and send the message to the determined user with UDP. For broadcast, we send the message to all online users after updating registry. In the multicast command, we send the message to the determined users. We exit the program with the exit command. Additionally, while we are sending the message, we add the username of the sender in front of the data to be sent. Therefore, in the listen side, we can understand who sent the message. Here, we separate the username of the sender and the data to be sent with a '~' character.

2) The list is in the following.

- The messaging could be mutual. In other words, each client can both send and receive messages.
- Instead of sending the messages with UDP, we could send the messages with TCP so that we could find out the packet losses and resend them again.
- Long messages could be sent by small size packets to prevent more losses.