Computer Engineering 4DN4

Lab #4

Online Group Chatting Network Application

April 8, 2018
Moshiur Howlader - 1316948
Ankur Bargotra – 1306226

This application was built off of the EchoClientServer_Thread.py code posted on the class website. This code had the threading logic to help manage handling of multiple users.

The data structures used for the Server Class of the applications were: a list containing `chat_room_list`, a list containing `multicast_socket_list`, a list of list containing a `replay_list`, a list for storing a set of TCP connections made from a client to server `thread_list`, and a list containing threads of multicast UDP connections required for the chat functionality to work, `multicast_thread_list`.

Note the TCP communications were ended by a delimiter of "÷".

The main changes added to the Server class were in functions: 1 - `connection_handler()`, 2 - `create_room_thread()`, 3 - `create_room()`, 4 - `delete_chat_room()`, 5 - `replay_chat()` and 6 - `save_chat_replay()`.

Function 1 handles the client inputs from the terminal. As specified from the document, the following commands were implemented: ***list***, ***create***, ***delete***, ***replay***, and ***bye***. The first four commands call functions 2, 3, 4, and 5 respectively to do the appropriate service for the client.

Function 2 creates a new thread for the chat room to exist.

Function 3 setup the UDP multicast socket on listen mode to listen in on the chat thread/room and to be able to continually keep track of what is being said on that group chat room.

Function 4 deletes the specified chat room.

Function 5 replays the chat room history of specified room from the `replay_list` list.

Function 6 is a helper function for function 3's infinite loop used to save the chat history into `replay_list`.

The main changes added to the Client class were in the functions: 6 – `main_menu_forever()`, 7 – `crds_connection()`, 8 – `chat_mode()`, 9 – `send_multi_UDP_chat()`. The helper functions were as follows: 10 - `recv_multi_UDP_chat()`, and 11 - `connection_receive()`.

Function 6 prompts the client a menu to access various commands such as ***connect***, ***name***, ***chat***, and ***exit***.

Function 7 prompts the client the CRDS menu after the client connected to this server correctly. Prompts commands such as: ***list***, ***create***, ***delete***, ***replay***, and ***bye***.

Function 8 manages the bookkeeping required to allow the client to go into the "chat mode". This mode is made possible by first obtaining the appropriate chat room information, creating the appropriate UDP multicast socket connection to connect to the chat room's IP address and port, then finally creating the 2 thread processes required to transmit (function 9) and receive (function 10) chat room messages through the UDP multicast socket for the given chat room.

Function 9 is a helper function to allow the given client to continually send messages to the chat.

Function 10 is a helper function to allow the given client to continually receive messages from the chat.

Function 11 is the function to manage the TCP connection from the server. We keep receiving data until a delimiter of "÷" is received.