COIS 4310H: Computer Networks
Assignment #1

RFC

Author: Sabrina Chapados-Wlodarczyk
Date: February 4, 2021
Version: 1.0

Summary of Application:

This application is designed to allow for communication between two individuals running the same client, using a text-based interface. Clients can send messages back and forth to each other through a connection to the server.

Architecture:

The application has two components: a server, and a client.

The server creates a socket, which is then bound to port 53943. It accepts connections from exactly two clients on the same port. It then receives and sends messages contained in packets through a TCP connection.

When at least one of the clients is no longer sending packets, the server closes the connection and the program exits.

The client program first prompts the user to input a username, which is then sent to the server and used to identify them to other users.

Then the user is prompted to send a message if it is their turn. If it is not their turn, then they need to wait while other users are sending messages (in round robin fashion).

Through the client, users have the following options (entered into the packet destination field — note that all input is case-sensitive):

- Send a text message to the other connected user (“all” or “username”)
- Request a list of all connected users (“who”)
- Disconnect from the server (“bye”)

When one of the users disconnects, the server sends a message to notify the other user, and then closes the connection.

Packet Structure:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
|      Source Client Name     |    Destination Client Name    |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
|                    Packet Sequence Number                   |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
|                       Version Number                        |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
|                           Verb                              |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
|                           Data                              |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Packet Header:

- Source Client Name: char[] (string), 50 bytes.
  Username of the client sending the message.

- Destination Client Name: char[] (string), 50 bytes.
  Username of the client receiving the message.

- Packet Sequence Number: int, 2 bytes.
  Number of packet in a sequence (if more than one packet is to be sent).

- Version Number: int, 2 bytes.
  Code version number (1 = Assignment #1, 2 = Assignment #2, etc.)

- Verb: int, 2 bytes.
  Tells the server what type of message it is receiving.

Data:

char[] (string), 256 bytes.
Contains the message to be relayed to other users.

Verbs:

0 - Messages from the server. Passes the turn to the receiving client, allowing them to send a message next.

1 - Login request. Passes a client’s username to the server; the server then notifies any other connected clients that the new user has joined the chat.

2 - Message to all. The server relays the message to all connected clients (other than the sender).

3 - Private message. The server relays the message to a specific user only.

4 - Who request. The server replies to the source client with a list of all the connected users.

5 - Quit. Notifies the server that the source client has disconnected. The server notifies other connected clients, then closes the connection.

6 - Wait. Message from the server, telling the client to wait until it is their turn to send messages.