Electrical and Electronics Engineering 121 Software Project

General Specifications

1. Your task is to design with the whole class a protocol for a peer-to-peer chat application. You are free to use any programming language you can utilize for this activity. However, support will mostly be given for people who use Python as the rest of the course utilizes it.
2. This will be a class-wide activity in which the specifics for the protocol will be designed by the class based on some rough specifications listed here.
3. For the communication medium of the application, your application will be using TCP sockets to transfer data from one instance of the application to the other. Format of the data sent through the TCP connection will be decided by the protocol design.
4. For the chat part of the application, users must be able to set a nickname for themselves which will reflect on other users in the network. Another feature of the chat part is that private messages can be sent from one user to another without them being directly connected to each other. Users must also be able to communicate with other nodes through another peer and be able to pass around messages sent by one peer to another peer available to its network.
5. Simulated weights must also be communicated at connection time. Originally, this would be the round trip time (ping) or the time it takes for the server to receive sent data and acknowledge its receipt to the sender. A node must maintain a list of nodes it can communicate to whether directly or not as well as which node it will use to communicate with an indirectly connected node on the list. A chat command should reveal this list with all the related weights. Indirectly connected nodes must be optimized to have the least amount of latency.
6. For checking, the instructor will draw a graph representing the connections and assign parts of the group their node. Due to the common communication protocol, different implementations should be able to understand each other and connect successfully as well as demonstrate that all the features work as intended. The group might want to build a protocol that allows for extension of commands to include additional features not part of the official documented protocol or allow non-complete implementations to still perform a minimal set of the protocol to be able to participate.

Grading System

25% Document for class-wide protocol implementation of the required specifications

25% Connection functionality and node distance listing

25% Public chatting and connection-disconnection notifications

25% Private messages and node renaming