COMP3825 Network/Info Assurance

Course Project

TA: Jibran Abbasi [jaabbasi@memphis.edu](mailto:jaabbasi@memphis.edu)

Office Hour: Tuesday 4:00 – 5:00 pm

<https://memphis.zoom.us/j/87368046196?pwd=RTZQS0U3VHpHa09rM1IxTTRFS2RGZz09>

This project intends you to design a simple, fully functional **“Client/Server Reliable Chat Application”** in your favorite programming language. The application should have following component.

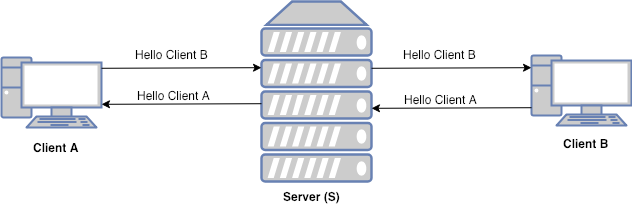


Fig: - Simple Client Server Chat Application Diagram

The application should implement the following functionalities:

* The application will contain a server and two or more than two clients (for chatting)
* Each client should be able to connect with the server. Once connected, the server will assign a unique identifier name to each of these clients.
* Once a new client (say Client A) is connected to the server, in response, the server will provide the list of all the available clients connected to it. In the simplest scenario, you can just have two clients connected to the server.
* Client (*i.e. Client A*) will use the identification name provided by the server to send the message to another client (say *Client B*).
  + You can decide the implementation you want for an identification number. (I would suggest a combination of ***random number + some Name***, by that way, a client can know with whom it is talking to).
* Once the server receives the message from a client (*Client A*), it will forward the message to the intended client (*Client B*) and vice-versa. (For simplicity: only implement for the case when both of the clients are available for the chat).
* A client can disconnect from the server by sending “.exit” message anytime. Server upon receiving “.exit” message from the client, will close the connection with it.

**Bonus (required if you have honors contract):**

* Implement chatting between multiple clients
* Implement simple encryption (e.g TLS) [1]

You can choose any programming language to develop a chat application. During your development/demonstration, you can run the clients and server on the same machine.

***Important:*** Please don’t forget to cite the sources you take help from.

**Deliverables:**

* **D1 Design Overview**: Submit a design document for the chat application by **11:59 pm, Mar 15th.** In addition to the text description, use class diagram, flow chart, or state transition diagram to explain your tentative implementation and design. (20 pts)
* **D2: Code of Framework:** Submit code framework for both client and server by **11:59 pm, April 5th.** It should have the major data structures and functions (APIs) defined based on the design document, and the connection between clients and server. (20 pts)
* **D3 Presentation and Demo, 11:59 pm April 18th** (30 pts). In class presentation, files submitted on Canvas.
* **D4 Final Report and Code**: Submit a final report (>= 3 pages) including design and evaluation results in the report, and final code showing (a) communication between the clients, (b) instructions on how to install and run the program, (c) screenshots of the input and output by **11:59 pm, April 21st** (30 pts)

**Reference & Help**

[1] “TLS/SSL wrapper for socket objects”, <https://docs.python.org/2/library/ssl.html>

[2] “Creating a simple chat Client/Server application” <http://pirate.shu.edu/~wachsmut/Teaching/CSAS2214/Virtual/Lectures/chat-client-server.html>

[3] “NETWORK PROGRAMMING - SERVER & CLIENT A : BASICS” https://bogotobogo.com/python/python\_network\_programming\_server\_client.php