

# CSCI 452/553: Network and Web Programming

## UDP Socket Programming Project

Due March 26<sup>th</sup>, 11:59:59 PM

- All development must be done with Github. Specifically, there must be at least one commit for every ten (or fewer) lines of code (excluding empty lines and lines with only comment). With each commit, the log must explain the purpose of the added/changed code. The Github repo must remain private before the deadline and public after the deadline.

- Submit a single plain text file named repo.txt at [http://hucs.dynu.net/lij/courses/submit\\_hw.html](http://hucs.dynu.net/lij/courses/submit_hw.html)

The text file should include only the URL of your Github repo.

- Grading  
All or significant portion of the points will be deducted at the discretion of the instructor if any of the above requirements are not met. Plagiarism will always lead to zero credit for this assignment and potentially for the whole course. No credits will be given if the repo is modified after the deadline. Other than that, the points will be given based on the correctness.

---

In this project, you will develop two network programs in the C/C++ one for client and one for server. You may base your program on the echo examples,

The client program reads input from user and performs the following actions repeatedly until directed to exit:

- If the user enters 's', it will ask user to enter a string. It then sends "xxx\n" to the server (where "xxx" is the string entered by the user, which could include any character except "\n"), and receives a message from the server and display the message on the screen.
- If the user enters 'q', it will exit.
- If the user enters anything else, it will ignore.
- If it receives a message from the server at any time, it will print the received message in a new line.
- It will automatically send "\n" to the server every 5 seconds.

The server program performs the following actions:

- Whenever a message that is not "\n" is received from a client, the server sends the same message back to all known active clients (i.e. the clients that the server has received messages from within the past 15 seconds), including the one the message is received from.

Clients should be invoked by the following command:

```
<client> <server IP> <server port>
```

Where <client> is the name of the client executable file name, <server IP> is the IP address of the server, <server port> is the UDP port of the server.

Note that multiple clients could be running at the same time, communicating with the same server.

The server should be invoked by the following command:

```
<server> <port>
```

Where <server> is the name of the server executable file name, <port> is the port the server listens to.