Assignment 3: Socket Programming

Goal of the Assignment: Create your own client server application using socket programming.

Extending Echo Client/Server for network analysis [Part 1]

1. **(15 M)** Create your own UDP echo client and server application to measure round trip time between client and server (similar to "ping" command)

   The client should create a UDP socket and send echo packets to server at a given interval, number of echo messages, and given packet size (use command line arguments). On reception of the packet, server should send the packet back to the client. The client on reception of the packet should calculate and display the round-trip time. To calculate the round-trip time, you can have the timestamp in the packet or/and use some unique identifier in the packet. You should also calculate and print the loss percentage at the end.

2. **(15 M)** Create an iperf like application using the above developed echo client and server program. Reduce the interval between two consecutive UDP echo packets generated by client to increase the number of echo packets sent from client for a given packet size. Calculate the throughput and average delay observed every one second. Plot the observed throughput and average delay vs time (1 second interval).

Extend Echo Client/Server and create your own client server application [Part 2]

(20 M) Add any two features to Echo Client/Server and demonstrate them. In the report, you must describe the new features with their benefit. **Significance of the features will impact the marks given.**

Making Echo Client/Server "protocol Independent" [Part 3]

(15 M) Revise echo client and server to be protocol independent **(support both IPv4 and IPv6).**
Hint 1: sockaddr is too small for sockaddr_in6. sockaddr_storage has enough size to support both sockaddr_in and sockaddr_in6. (You will see this in server-side program.)
Hint 2: integrate getaddrinfo to avoid typing IPv6 address on your CLI
Hint 3: you may use hostname (IPv4: "localhost", IPv6: "ip6-localhost" address to develop / demonstrate the software on ubuntu. They're written in "/etc/hosts".

Instructions for Implementation:

- You may choose any programing language (C, JAVA, Python, etc.)
- The software must be based on Socket Programming.
- Wrappers of API must not be used (messaging etc). Use send/recv or read/write using TCP/UDP socket.
- Keep the record of Reference.

Deliverables in a tar ball on GC:

1. A report detailing your implementation detail and the results.
   The core idea of your answer to each question. Better visibility

like screenshot of application will be appreciated. One single report file (<your roll no>_<Name>_<Assignment3>.pdf for all three parts, include the code in the report as well.

2. Screenshots of packet capture using Wireshark during your experiments.
3. Screenshots showing the working of your code.
4. All source codes and README as a separate file so that TAs and instructor can compile source code and execute the binary anytime.
5. Submit all files in a single zip file named as ,<your roll no>_<Name>_<Assignment3>.zip

Note: Plagiarism check will be done on your submitted code. This assignment will be evaluated through presentation. In the presentation, you are expected to explain and demonstrate the applications you built.