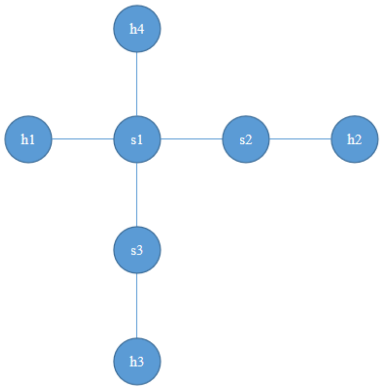
Lab 3: Socket Programming for Chatting Room with Multiple Users

Questions

This lab aims to help you get familiar with socket programming, which is a commonly used tool for messaging and file transmission in internet. In this lab, you will build two chatting rooms supporting multiple users with different methods. You need to program in C++ language. You can refer to https://beej.us/guide/bgnet/ for more useful information.

1. Chatting room with multiple users: client-server model with TCP protocol (50 points)
   Requirements:
   a) Use Mininet to build the following topology, which contains 4 hosts. (You may reuse the codes in lab2.) Pick one host as the server and others as clients.



   b) The workflow of this chatting room is as follows: Each client can always input a message into its terminal to send it to the server. The server is responsible for receiving the message from one client and then distributing the message to other clients, such that all the messages from clients can be displayed on each client's terminal. (In Mininet, the command to open the terminal of host h1 is 'xterm h1'.)

   c) You are supposed to build the above mentioned chatting room using socket programming with TCP protocol. You need to write two cpp files, namely, server.cpp and client.cpp. And the two files should be complied by following commands.

   g++ server.cpp –o server and g++ client.cpp -o client

   And you are supposed to run the executable files on terminals of hosts as

   ./server and ./client

   Please obey the above mentioned instructions carefully because we will test your code with exactly the same procedures.

2. Chatting room with multiple users: client-only model with UDP protocol (50 points)

   Requirements:

   a) Use Mininet to build the same topology as in Problem 1.

   b) The workflow of this chatting room is as follows: Each client can send the message to the broadcast address. At the same time, they also continuously listen to the messages from the broadcast address. In such way, each client can transmit/receive messages to/from other clients without the help of a server. All the messages from clients should be displayed on each client's terminal.

   c) You are supposed to build the above mentioned chatting room using socket programming with UDP protocol. You need to write one cpp file, udpclient.cpp. It should be complied by following command.

```
g++ udpclient.cpp -o udpclient
```

   And you are supposed to run the executable files on terminals of hosts as

```
./udpclient
```

   Please obey the above mentioned instructions carefully because we will test your code with exactly the same procedures.

3. **Bonus question：** For each client that you built in the above Question 2, it will receive the messages sent by itself from the broadcast address. Can you filter out these messages, such that each client's terminal will not display the message sent by the client itself? Bonus will be offered, if you could realize this function.

Your final submission .zip file should include a pdf report, server.cpp, client.cpp and udpclient.cpp. Please put the screenshots of hosts' terminals in your report.