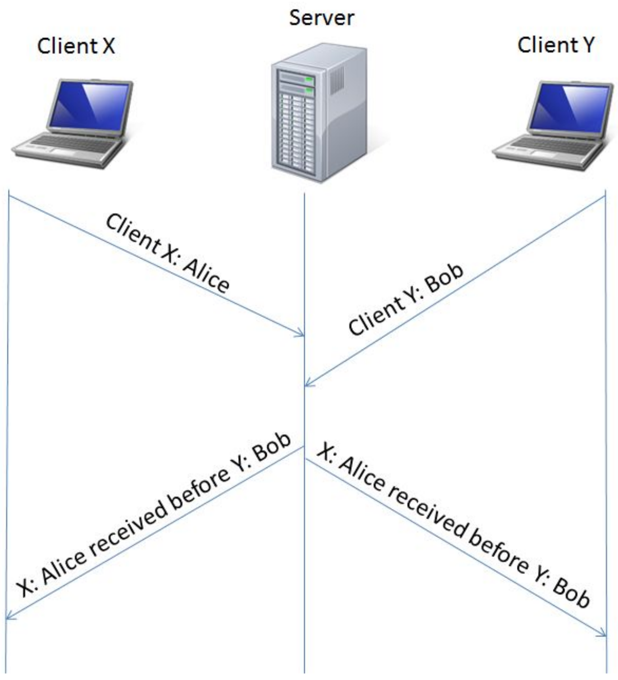
CST 311

Instructor: Michael McCann

Programming Assignment 2: Socket Programming

In this assignment, you'll write a server and client program. There are two clients, X and Y (both essentially identical) that will communicate with a server. Clients X and Y will each open a TCP socket to your server and send a message to your server. The message contains the name of the client followed by your name (e.g., "Client X: Alice", "Client Y: Bob").

The server will accept connections from both clients and after it has received messages from both X and Y will print their messages and then send an acknowledgment back to your clients. The acknowledgment from the server should contain the sequence in which the client messages were received ("X: Alice received before Y: Bob", or "Y: Bob received before X: Alice"). After the server sends out this message it should output a message saying - "Sent acknowledgment to both X and Y". Your server can then terminate. Once your clients receive the message from the server, they should print the message that they sent to the server, followed by the reply received from the server. The following figure explains the problem.


Client X
Server
Client Y
Client X: Alice
Client Y: Bob
X: Alice received before Y: Bob
X: Alice received before Y: Bob

Your program should print out the messages sent by the client at both the client and server and vice versa for messages sent by the server.

Extra Credit: (25%) The extra credit part of this assignment is mainly for **fun** and is Challenging.

> You can modify the above server-client to create a simple chat service. Clients X and Y can only chat through the server. For example, every message that client X sends to the server, the server relays to client Y and vice versa. When a client (say X) wants to exit the chat service it sends a “Bye” message. When a server sees a “Bye” message, it relays this message to Y and then terminates the connection to both clients. Each client (say X) should output the messages sent by it and those received from Y. As this is a chat service the number/content of messages exchanged is not fixed. So your clients should have the capability to accept inputs (which are the content of the messages) from the keyboard.

Write your programs in Python 2.7 and execute them on your mininet virtual machine.

What to hand in?

1. Submit code (as files in a .zip or tar ball) with inline documentation
2. A design document outlining the design decisions you made
3. Sample outputs (screenshots) which show that you program works correctly
4. A brief description of cases where you code might fail and possible ways of improving your program

Grading Criteria

- In-line documentation - 20 points
- Code compiles and executes correctly – 60 points
- Design document – 20 points