CS304– Computer Network

Instructor : Shashi Shekhar Jha (shashi@iitrpr.ac.in)

Lab Assignment - 1 | Due on 17/02/2020 2400 Hrs | 50 Marks

Submission Instructions:

All submission is through google classroom in one zip file. In case you face any trouble with the submission, either contact the TAs or you can post your queries in the classroom.

Your submission must be your original work. Do not indulge in any kind of plagiarism or copying. Abide by the honour and integrity code to do your assignment. It is okay to discuss the problems with your peers but all implementations should be your own work.

As mentioned in the class, late submissions will attract penalties.

Penalty Policy: There will be a penalty of 5% for every 24 Hr delay in the submission. E.g. for first 24 Hr delay the penalty will be 5%, for submission with a delay of >24 Hr and < 48 Hr, the penalty will be 10% and so on.

You submission must include:

- A legible PDF document with all your answers to the assignment problems, stating the reasoning and output.
- A folder named as 'code' containing the scripts for the assignment along with the other necessary files to run yourcode.
- A README file explaining how to execute your code.

Naming Convention:

Name the ZIP file submission as follows:

`YourName_rollnumber_Assignmentnumber.zip`

E.g. if your name is ABC, roll number is 2017csx1234 and submission is for lab1 then you should name the zip file as: `ABC_2017csx1234_lab1.zip`

Section 1: Common Network Utilities

================================

Execute and understand the usage of the following networking commands/tools that are used to debug and monitor the status of networks. For each of the following commands, describe their use along with their outputs with various options available with these commands (such as for Ping command, few options are $ping -a <target> or $ping -t <target>). Hint: Look into the man pages (e.g. man NSlookup) of these commands on a Linux terminal. [12 point]

1. Ping
2. Traceroute
3. NSlookup
4. NetStat
5. Ipconfig/ ifconfig
6. Hostname

Use the following targets for the first three commands mentioned above.

1. iitrpr.ac.in
2. google.com
3. google.co.in
4. gmail.com
5. facebook.com
6. wikipedia.org
7. india.gov.in
8. nationalgeographic.com
9. nkn.gov.in
10. irctc.co.in

Mention the following in the report document:

a) Analyze the output of Ping command for each of the the target site and report the maximum, minimum and average round-trip time (RTT). RTT is the time taken by the request to travel from your machine to the target site plus the time taken by the response from target to reach back. [2 point]

b) Report the packet loss percentage for each site. [1 point]

c) What is the default size of Ping packet sent to the target site and the packet size received. Change the default ping packet size to 100 bytes and again report the RTT and packet loss percentage. [2 point]

d) If you have observed the outputs of traceroute for different target sites, you must have seen that upto a few levels the IP addresses on the route of trace packets are same and then the packets take different routes to the servers of various target sites. Why is so? also, can you mention where do the common IP addresses belong to? [5 point]

Section 2: Sockets

===============

This section is related to socket programming using python language.
You can find more about Sockets in Python at the following link:
https://docs.python.org/3/howto/sockets.html
There are also various resources available over the Internet to help you with an understanding of the socket programming.

What are Sockets?

Typically, the Sockets are the endpoints (in software) of a bidirectional (full-duplex) communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different machines placed across the globe.

Sockets may be implemented over a number of different channel types: Unix domain sockets, TCP, UDP, and so on. The socket library provides specific classes for handling the common transports as well as a generic interface for handling the rest.

1. In this part of the assignment, you will be writing two programs viz. `server.py` and `client.py`. A Client always initiates connection with the server program and Server always responds to the requests of the clients. The server you are going to create will act as an authentication server to provide access to the students enrolled in CS304 course. The `login_credentials.csv` file consists of the login information of the users which needs to be available with the server in order to authenticate requests for access from client hosts. Run `client.py` and `server.py` at two different systems/hosts/PCs.

Your client and server need to achieve the following requirements:

I. Your program for client needs to take two arguments that specify the name of server and the port that it is trying to connect to. Your program for server needs to take an argument that specifies the port that it is listening to. [2 point]

II. Your server will start first and keep listening to the specified port. Your client will connect to the port that your server is listening to, and a socket between your client and server is constructed. Show a message at both server and client ends that connection is successful. [1 point]

III. Your client will first prompt a welcome message that asks the user to enter a username using the keyboard. This username will then be sent to the server. Then, your server, after receiving the username from your client, will send an acknowledgment message to the client. [2 point]

IV. Your client, after receiving the acknowledgment message from your server, will prompt a message that asks the user to enter the corresponding password. This password will then be sent to the server. Then, your server, after receiving the password from your client, will verify the received pair of username and password against the list of legitimate pairs. If the result is positive, the server will send a success message to the client. If the result is negative, the server will send a failure message to the client. [3 point]

V. Your client, after receiving the result message, will print out the result and close the socket. Your server will close the socket following the client, and keep listening for the next client request. [2 point]

2. Make your server program to respond to more than one client at a time. [3 point]

3. Run the client and server programs at two different machines/hosts. Find the bandwidth of the medium available for your server and client connection. Hint: Think in terms of the data rate at which you are able to send and receive information. [2 point]

4. Using the skills acquired after solving the previous questions on socket programming for communicating between a sender and a receiver, let us now create a distributed authentication system. In such a system, a single server may not have the full information to perform authentication. Instead, the login information is distributed among three record_keeping hosts (A, B, C) in the network. Distribute the data in

login_credentials.csv in a balanced manner to the three hosts, A, B, and C. Further, you want to provide exclusive access to the students of CS304 with 80% or more attendance. For this, there is another host (D) with attendance records of the users available in the file attendance_percentage.csv. Here, your original server.py program now needs to be altered to have the functionalities of both server and client. After collecting the username and password pair from the client, the server needs to contact the record_keeping hosts A, B and C to validate the authentication. Further, the server also needs to contact the attendance host D in order to validate the attendance record of the user before sending authentication successful/failure information to the Client. [10 point]

The information regarding the hosts A, B, C and D must be stored in a routing table a file named <host_name>.rtl. Also allow a special input word by the user, eg. EXIT, to terminate any host. Print the messages for all the communication exchanges among at all the hosts. [1+1 =2 point]

Your implementation should be such that any configuration of a 4 node network could be created by changing the routing table of the hosts. [1 point]