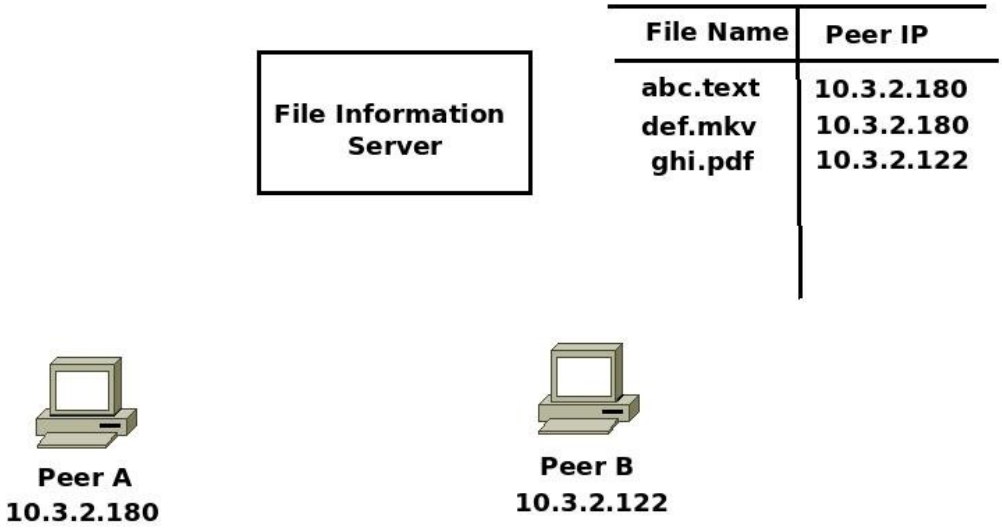
CS 39006: Networks Lab
Assignment 5: Peer to Peer File Sharing Through Socket Programming (C/C++)

Date: 21st March, 2017

Assignment Statement:

In a Peer-to-Peer system, many computers are interconnected through a network to share information and resources among themselves either through a centralized server or without a centralized system, like the interconnected hub that we use in DC++. In a client-server model, server has the information about all clients and the clients can contact the server for the file information of clients, and then contact the corresponding clients to download the required files.

In this assignment you'll be implementing a small scale peer-to-peer communication system for file sharing, using socket programming API's in C or C++. There will be a file information server that contains the peer information, and the files available to every peer. The system architecture looks like following:



The assignment has two parts. In the first part, you need to implement the system for two peers and a single file information server. The design details are as follows.

1. You need to implement a File Information Server (FIS). The FIS maintains a table that has information about all the peers' IP addresses and the files available to those peers.
2. You also need to implement server and client processes at the individual peers. The communication protocol between the peers and between a peer and the FIS is as follows.

a. Peer A wants to download a file named abc.pdf. The file is available at Peer B.
b. Peer A communicates with the FIS to get the details of Peer B
c. Then Peer A communicates with Peer B to download the file abc.pdf.
d. **The communication between a peer and the FIS is through Datagram sockets.**
e. **The communication between two peers is through stream sockets.**

In the second part of the assignment, you need to extend the system for N number of peers, where N can be any number. In this scenario, a single peer can have multiple files. Note that, the FIS can receive simultaneous requests from multiple peers. Similarly, a single peer may receive file download requests from multiple other peers. Your system should be scalable enough to handle such complexities. (Note: Check select() system call for the socket API to find out how to handle multiple requests simultaneously).

Submission instructions:

- **FIS_Server.cpp** – the server code for FIS
- **Peer_Server.cpp –** the server code for peers (for downloading files)
- **Peer_Client.cpp –** the client code for peers (this includes two clients – one datagram client to get the information from FIS, and one stream client to download the file from other peers)

Notes:
Use port numbers in the range from 10000 to 15000.
Implementation to be done in C or C++
The implementations have to be system independent.

Submission Instructions:

You need to prepare a report that will contain the followings.

3. Steps followed in executing the experiments.
4. Observations from the experiments.
5. The scripts corresponding to topology construction, controller, L2 forwarding and L3 routing.
6. Intuitive justification behind the observations.

You need to submit the report and relevant scripts (source files) in a single compressed (tar.gz) file. Rename the compressed file as Assignment_4_Roll1_Roll2.tar.gz, where Roll1 and Roll2 are the roll numbers of the two members in the group. Submit the compressed file through Moodle by the submission deadline. The submission deadline is: **March 28, 2017 02:00 PM.** Please note that this is a strict deadline and no extension will be granted.

Please note that your submission will be awarded zero marks without further

consideration, if it is find to be copied. In such cases, all the submissions will be treated equally, without any discrimination to figure out who has copied from whom.