Homework 3 - P2P File Sharing System

Due: April 21 , 2020 23:55

You can do this homework in a team of up to 2 students. Include the names of all your team members in the beginning of your program. Any form of cheating will not be tolerated.

In this homework you will develop a peer-to-peer (P2P) file sharing network application in Java (or Python) with GUI[1]. The application should allow users to share and download files from each other.

In general, the system should work as follows. In the system, one designated machine, let's call it FileTracker (FT), should always be on and waiting for others' (peers) requests. FT should contain *information about the files* which are shared by *online* peers and *information about the online peers*. If a peer A wants to download a file X, A asks FT to send information about the online peers that share X. If there exist peers that share X, FT sends information about these peers and about files to A. Note that different files with the same name may exist (difference could be in size, type, etc.). A connects to one of the peers that it has received from FT and *may* download X from that peer.

The rest explains how the system should work in detail. See Figure 1 for illustration.

- The IP address of FT and the port number on which the application is running should be public, i.e., users should know the IP address and port number of FT.
- When a peer A connects to FT, A should automatically send information about its shared files (up to 5 files) as a list of records to FT in the following format: <file name, file type (e.g., text, jpg, etc), file size, file last modified date (DD/MM/YY), IP address, port number>. A should send "HELLO" and receive "HI" before sending the information.
- If A does not send any file information to FT while joing the system, i.e., if A does not share any file, then FT should not accept A. FT should not respond to A.
- *Only* accepted peers should be able to use the services offered by this system.
- When A wants to download a file with the name "File Name", A requests the file from FT by sending "SEARCH: " + "File Name".
- When FT receives "SEARCH: " + "File Name", it tries to find the file in a *hash table* where 'key' is filename and 'value' is a list which contains records of this format: <file type, file size, file last modified date (DD/MM/YY), IP address, port number>.
  - If FT finds the file, it should send "FOUND: " + list of records.
  - If FT does not find the file, it should send "NOT FOUND".
- After receiving a list of records, A should choose one of the peers (records) from the list, say B, and connect to B (using IP and port number) to request and download the file. For that A should send "DOWNLOAD: " + "FileName, type, size" to B.
- When B receives a "DOWNLOAD" message from A, it should send "FILE: "+file to A.
- When A wants to leave the system, A should notify FT about this so that FT can update the list of online users. A should send "BYE" to FT to do so.

[1] A sample Java program with GUI will be given on the course's moodle page

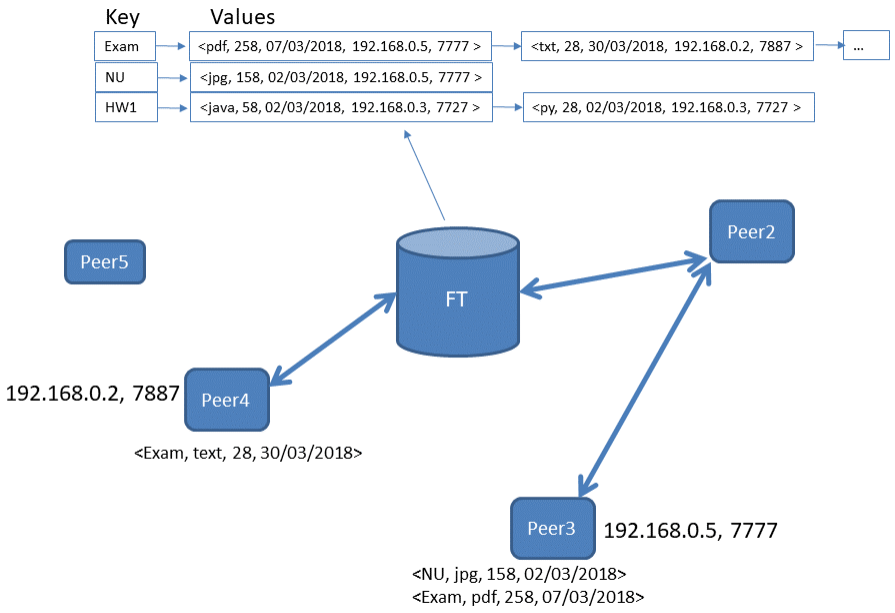

Figure 1: Illustration of P2P File Sharing System.