Gnutells style Peer-to-Peer File Sharing

Abdullah Ejaz

May 3, 2018

Design Document

1 GitHub Link

Click here to go to the gitHub repository of the project.

2 Overview

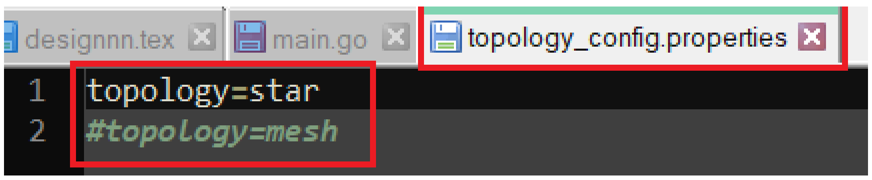
In this project a Gnutella style Peer-to-Peer file sharing network has been implemented using java and socket. The architecture of the network is completely decentralized meaning there is no centralized server. All the peers themselves behave both as a server and as a client.

- As a client, peers provide interfaces through which users can issue queries and view search results.
- As a server, it accepts queries from other peers, checks for matches against its local data set, and responds with corresponding results.

Since there is no central server, if a peer has to search for a file, it will be done in a distributed manner. Each peer will send a broadcast message to its neighbor peers and then they will broadcast it to theirs with the name of file to be searched in the search query. If the file will be there on the network with any peer, that peer will send a reply with a hit query message to the main searcher peer. The peer then can directly communicate with the node having the file and download it to its local directory.

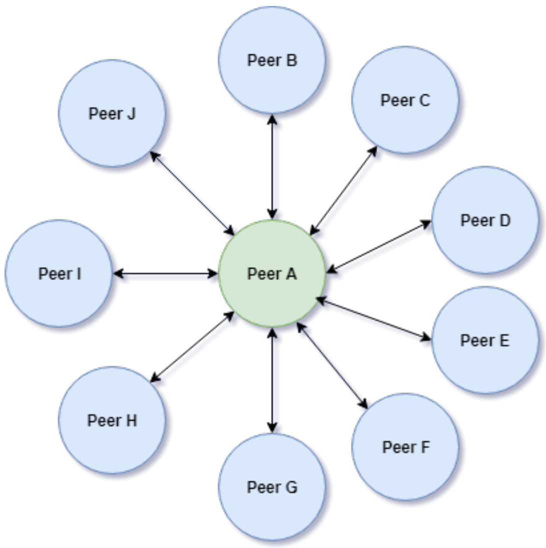
3 Network Topology

In this project, the gnutella style peer-to-peer file sharing has two different network topology which are Star and Mesh. The program uses a topology configuration file to decide on which topology network will be running on, which looks like this.



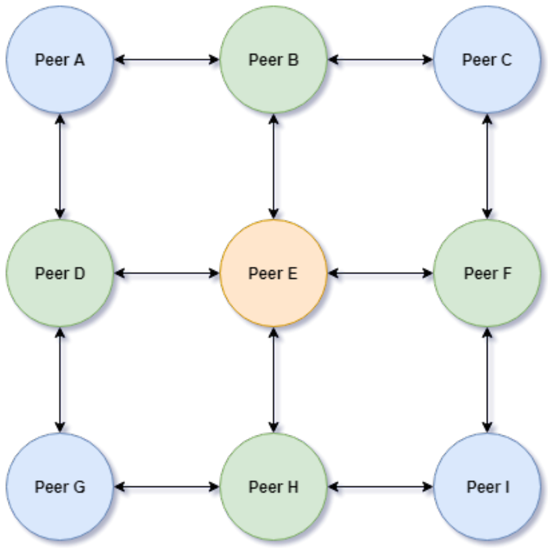
```
topology=star
#topology=mesh
```

3.1 Star Topology



STAR TOPOLOGY

3.2 Mesh Topology



MESH TOPOLOGY

4 Architecture

4.1 Peers

As we do not have any centralized server in the program. Peer behaves in both the ways as a server and as a peer.Every peer has has some functionality which includes searching for a file, downloading a file etc which we will see in the further section of the report. The interfaces provided by the peers are as follows.

- **configFileRead(peerId, topology)**
  - This method is the very starting of the program. As soon as the program starts running it will read the configuration file to check the network topology. There are two different topology for the network as discussed above. The program will read the file named `topology_config` in the root folder of the project and select the topology. User can change the topology by editing the configuration. If the user wants to run the mesh topology then he/she should prefix the line "topology=star" with a hash symbol to comment it out and then uncomment the "topology=mesh" line. If in case both the lines in the configuration files are commented, then the system would start on the default topology that is being set as "STAR".

- **availableFilesAtPeers()**
  - Every peer has its own dedicated local directory in which there are all the files that peer owns. This method gives us the list of the files present in the local directory of the peer and also the list of the files peer downloaded from other peers.

- **registration(fileName)**
  - This method helps a peers to register its files with the network. This methods takes the fileName as the parameter. But this method only registers one file at a time. If the peer wants to register all the files of the local directory in a single go then the next explained method would be used.

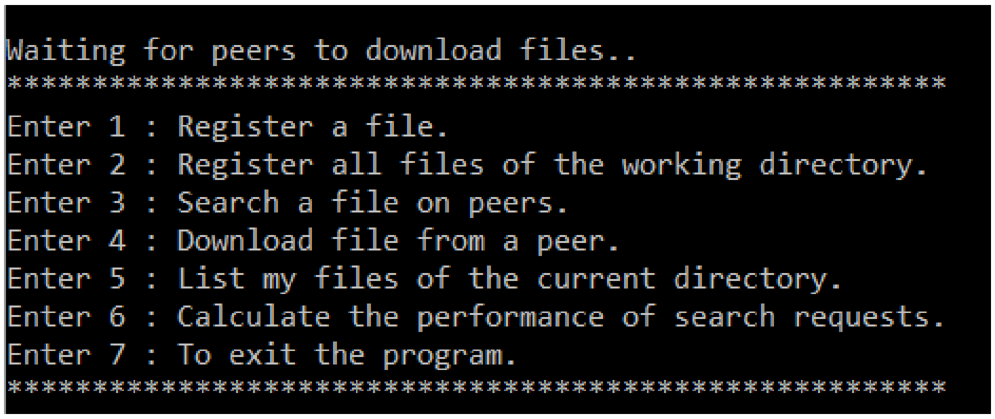
- **registration_AllFiles()**
  - This method helps a peer to register all of its files with the network in a single go.

- **downloadFromPeer()**
  - After a peer knows about the location of the file, the next function could be to download it. This method of the MainServices class help the peers to download a file directly from other online peers on the network.

- **broadcast(message)**
  - This method as the name suggest helps broadcasting the message among the neighboring peer about the search of a file. If a file has been found at any peer then this will send a hit query to the main searcher.

- **checkFileValidity()**
  - This method keeps on running in the background to check the validity of the files in the peer folders. If a file is corrupt, then it will delete the invalid file.

4.2 Functionality of Peers



```
Waiting for peers to download files..
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
```

- A peer can register a file on the network by choosing the option 1.
- A peer can register all of its files on the network by choosing option 2.
- A peer can search for a file on the network.
- A peer can download a file if it is available at any other peer.
- A peer can print the list of all the files it has in the local folder
- A peer can do multiple search requests to calculate the performance.

5 Future Enhancements

- A GUI would really help in executing the program well.
- Dynamic allocation of the ports would be good in terms of avoiding overlaps.