Design Doc

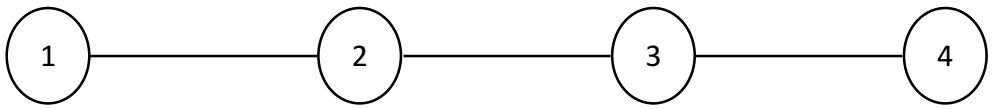
We have used the Java Sockets to implement GNutella type network. We have kept the TTL to 7 by default which will decrement when it hops from one hop to another .

Configuration file-

We have kept list of neighbors in a separate like ***portNumberconfig.file***

Below configuration for linear & star topology, where ip is the ip address of neighbor.

Linear Topology



File Contents:
1config.file
ip:2

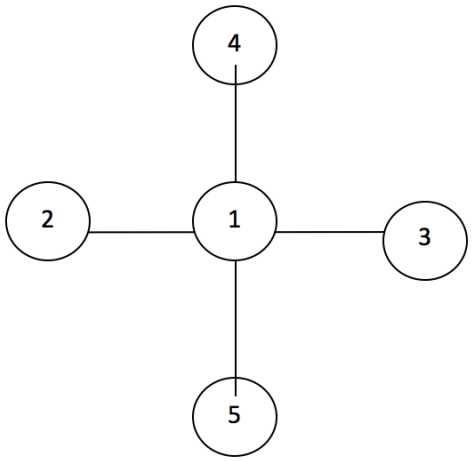
2config.file
ip:1
ip:3

3config.file
ip:2
ip:4

4config.file
ip:3

Star Topology

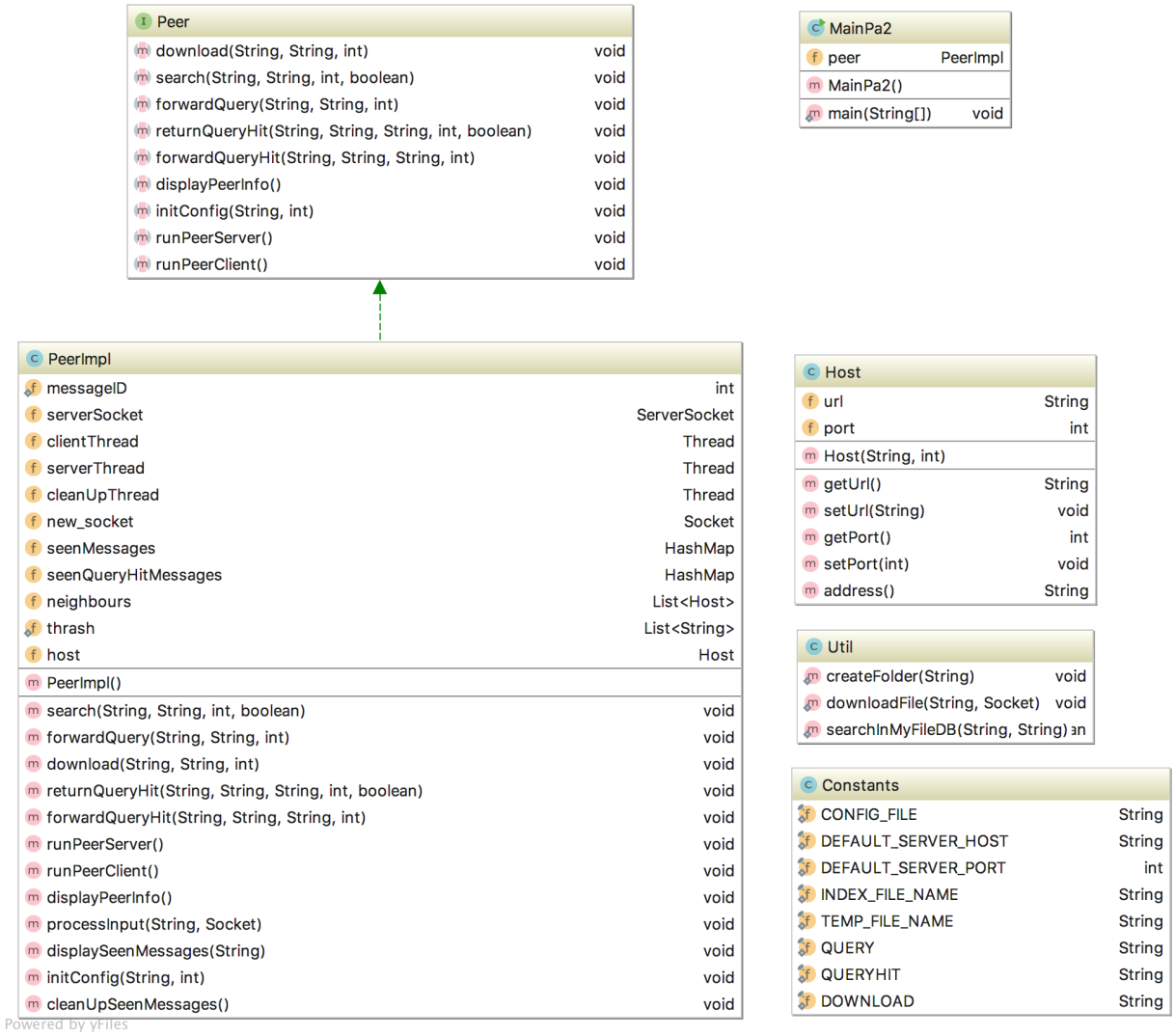
Peer Configuration File for star topology

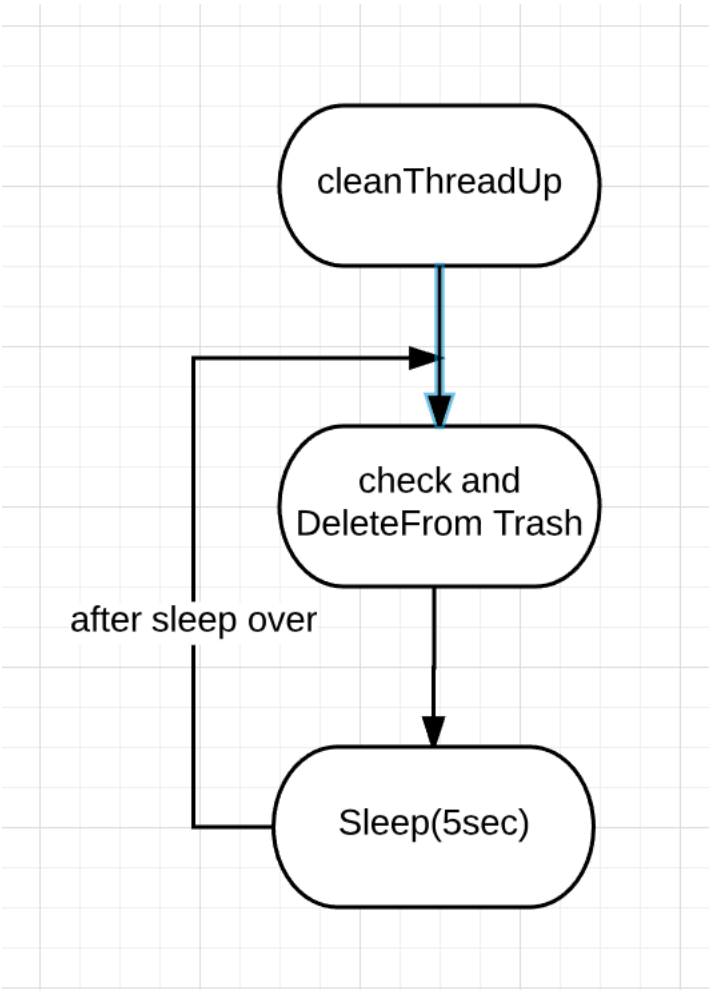


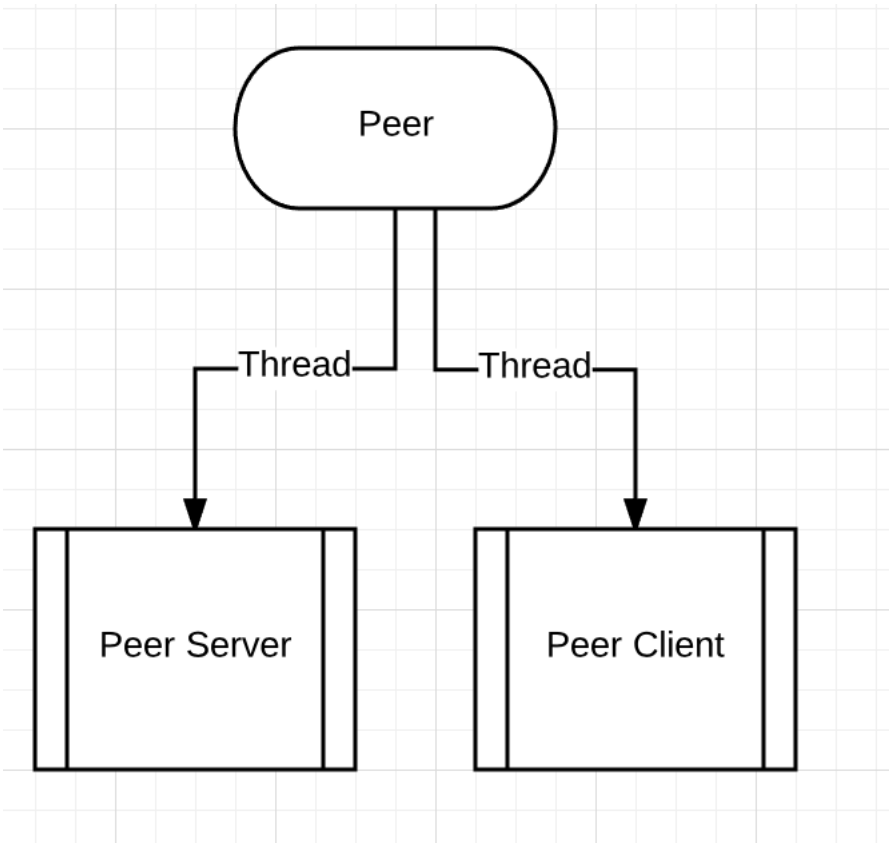
File Contents:
1config.file
ip:4
ip:3
ip:2
ip:5

2config.file
ip:1
3config.file
ip:1
4config.file
ip:1
5config.file
ip:1

Class Diagram :




cleanThreadUp
check and
DeleteFrom Trash
after sleep over
Sleep(5sec)



Manual text

```
Steps to run our program :

1. Place the shared shell script "run.sh" , shared jar file and corresponding
configuration file in a seperate folder for each peer.
   For example :
   if I want to create a peers running on port 52001, I will create a folders(any
name) and place "run.sh" , jar file and its config file "52001config.file" in it.
2. Run the shell script (./run.sh)
3. Default config? Y/N, Say N
4. Enter ipaddress of the system where peer is running (if it this system itself,
enter "localhost")
5. Enter port where this peer has to run.. (I will say 52001 if I have placed
52001config.file)
6. later select options accordingly
7. If you download any files it will be stored in "sharedFolder(ip:port)". in my case
"sharedFolderlocalhost:52001"
8. Any file shared by this peer should also be stored in this folder.. this folder
will be created on peer's bootup
```

Example tree structure -

```
23:40:41 macOS_ ⚡  tree
.
├── 5555config.file
├── FileSharingSystem-1.0-SNAPSHOT.jar
└── run.sh

0 directories, 3 files
```