

CS550 Programming Assignment 3 (PA#3)

Maintaining File Consistency in the Hierarchical Gnutella-style P2P File Sharing System

User Manual

Configuring the Network topology in the config.properties file:

As we are not implementing a dynamic Gnutella network, we initialize and define the topology structure in the config.properties file.

There are two types of peers in the system:

1. A Super Peer which is acting like a proxy index server for the leaf nodes.
2. A leaf node is the peer which has only one connection which is connected to the super peer.

Step 1: To initialize peers in the network

Leaf node:

A leaf node has the following details.

One needs to configure the IP address, a unique port number and a path to a shared directory.

```
peerid.2.ip=127.0.0.1
peerid.2.port=2000
peerid.2.shareDir=\\src\\peer2
peerid.3.ip=127.0.0.1
peerid.3.port=3000
peerid.3.shareDir=\\src\\peer3
peerid.4.ip=127.0.0.1
peerid.4.port=4000
peerid.4.shareDir=\\src\\peer4
```

Super Peer:

A super peer has the same details as the leaf node, except there is no shared directory.

```
peerid.1.ip=127.0.0.1
peerid.1.port=1000
```

Step 2: To initialize the network topology

We need to set the neighbours of the peers.

In the below example, peer2, peer3 and peer4 have only one connection to peer1,

i.e. Peer1 is a super peer and the rest peer2 3 and 4 are the leaf nodes.

Peer 1 has multiple connections, i.e. it is connected to its leaf nodes as well as other super peers in the network.

```
peerid.1.neighbors=peerid.3,peerid.2,peerid.4,peerid.5,peerid.37
peerid.2.neighbors=peerid.1
peerid.3.neighbors=peerid.1
peerid.4.neighbors=peerid.1
peerid.5.neighbors=peerid.6,peerid.7,peerid.8,peerid.1,peerid.9
```

```
peerid.6.neighbors=peerid.5  
peerid.7.neighbors=peerid.5  
peerid.8.neighbors=peerid.5
```

Setting Approach and Time to Refresh:

File Consistency is implemented using two approaches, one is push and the other is pull. One can use one or the other and not both, as both are basically two approaches for file consistency. Comment the pull approach by prefixing '#' if you want to use push approach and vice versa.

```
approach=push  
#approach=pull
```

One can also configure the Time to refresh value as mentioned in the below property.

```
ttr=100
```

Please follow the following steps to execute the Assignment:

Step 1: Configure the configuration properties file as per the network requirements.

Step 2: Open command prompt and navigate to the directory which contains all the source files of the server. Execute the following command.

```
javac *.java
```

Step 3: Execute the following command

```
java Peer
```

Step 3a: In the Peer window Enter the peer id as mentioned in the configuration file.

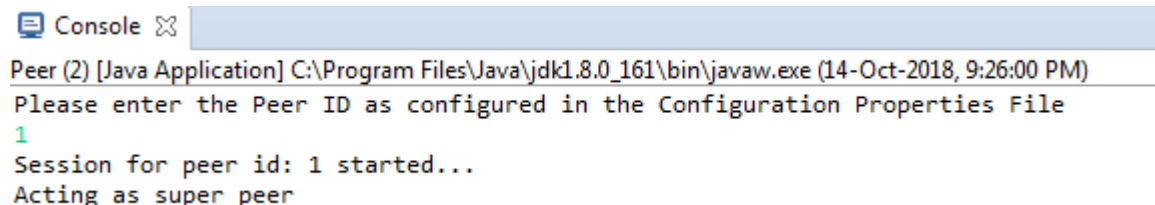


Figure 1: Screenshot of the Super Peer

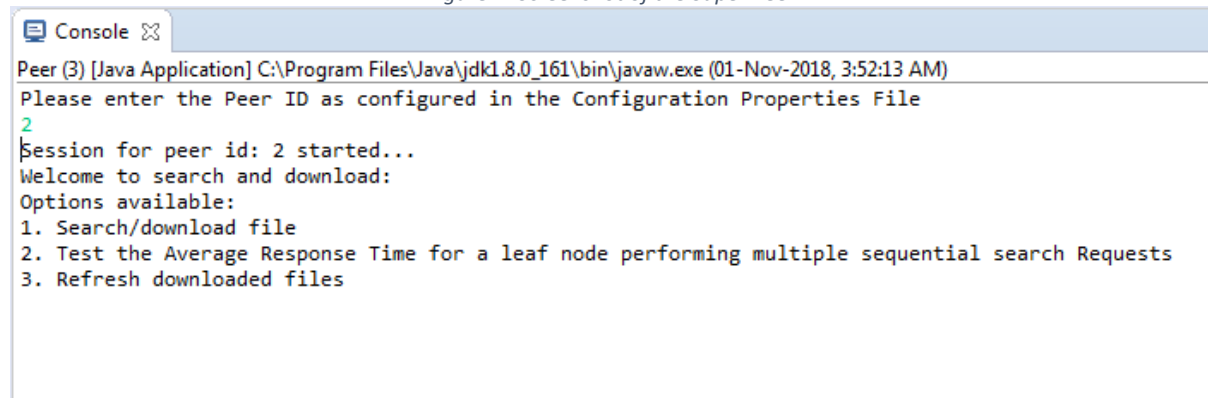
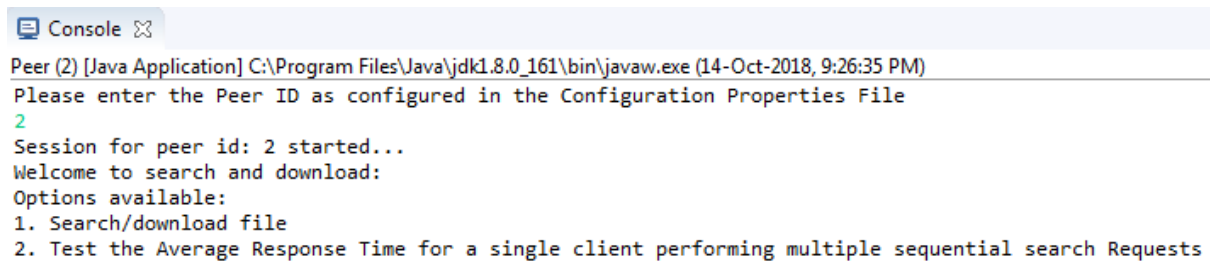
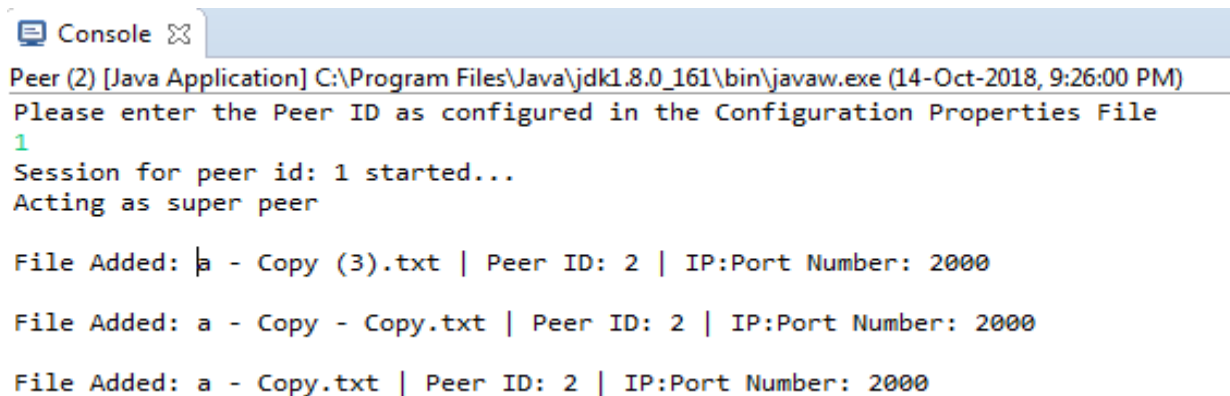


Figure 2: Screenshot of a Leaf node (Pull approach)



```
Console
Peer (2) [Java Application] C:\Program Files\Java\jdk1.8.0_161\bin\javaw.exe (14-Oct-2018, 9:26:35 PM)
Please enter the Peer ID as configured in the Configuration Properties File
2
Session for peer id: 2 started...
Welcome to search and download:
Options available:
1. Search/download file
2. Test the Average Response Time for a single client performing multiple sequential search Requests
```

Figure 3: Screenshot of a Leaf node (Push approach)



```
Console
Peer (2) [Java Application] C:\Program Files\Java\jdk1.8.0_161\bin\javaw.exe (14-Oct-2018, 9:26:00 PM)
Please enter the Peer ID as configured in the Configuration Properties File
1
Session for peer id: 1 started...
Acting as super peer

File Added: a - Copy (3).txt | Peer ID: 2 | IP:Port Number: 2000
File Added: a - Copy - Copy.txt | Peer ID: 2 | IP:Port Number: 2000
File Added: a - Copy.txt | Peer ID: 2 | IP:Port Number: 2000
```

Figure 4: Screenshot of Super Peer after the leaf node is registered.

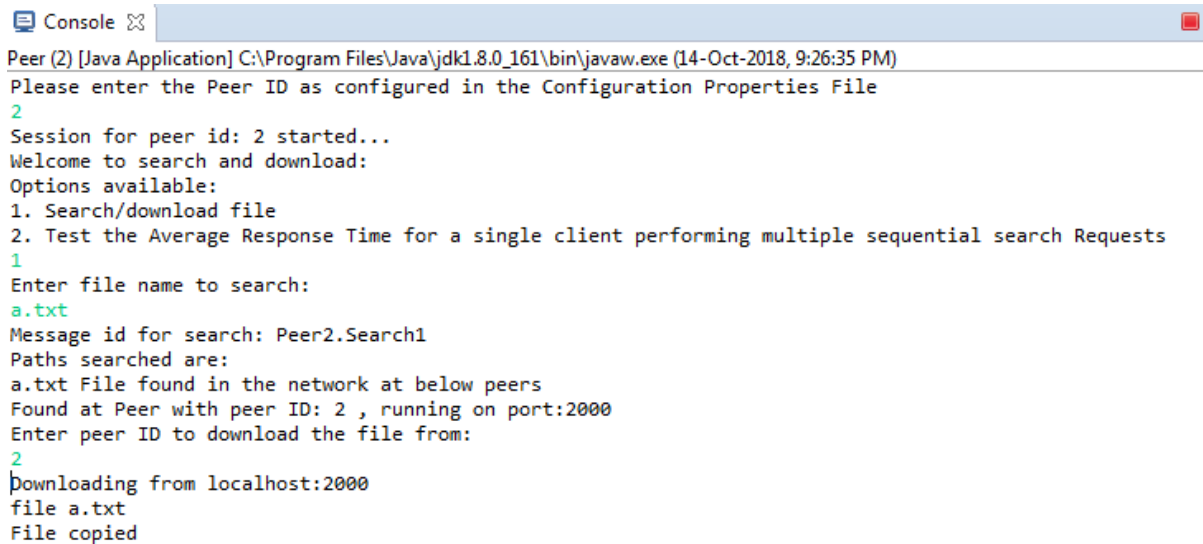
Repeat Step 3 for all the peers initialized in the configuration in different command prompts.

To download a file:

Precondition: The network should be initialized and super peers and leaf nodes should be running.

In the Leaf node window,

- Press 1 to Start search and Download a file
- Enter the file name which the other client has in its shared directory. Eg.: file1.txt
- The program will display the traversed path used to get the peer information.
- The program will give a list of peers that contain the searched file.
- Enter the peer id.
- The file has been downloaded.



```
Peer (2) [Java Application] C:\Program Files\Java\jdk1.8.0_161\bin\javaw.exe (14-Oct-2018, 9:26:35 PM)
Please enter the Peer ID as configured in the Configuration Properties File
2
Session for peer id: 2 started...
Welcome to search and download:
Options available:
1. Search/download file
2. Test the Average Response Time for a single client performing multiple sequential search Requests
1
Enter file name to search:
a.txt
Message id for search: Peer2.Search1
Paths searched are:
a.txt File found in the network at below peers
Found at Peer with peer ID: 2 , running on port:2000
Enter peer ID to download the file from:
2
Downloading from localhost:2000
file a.txt
File copied
```

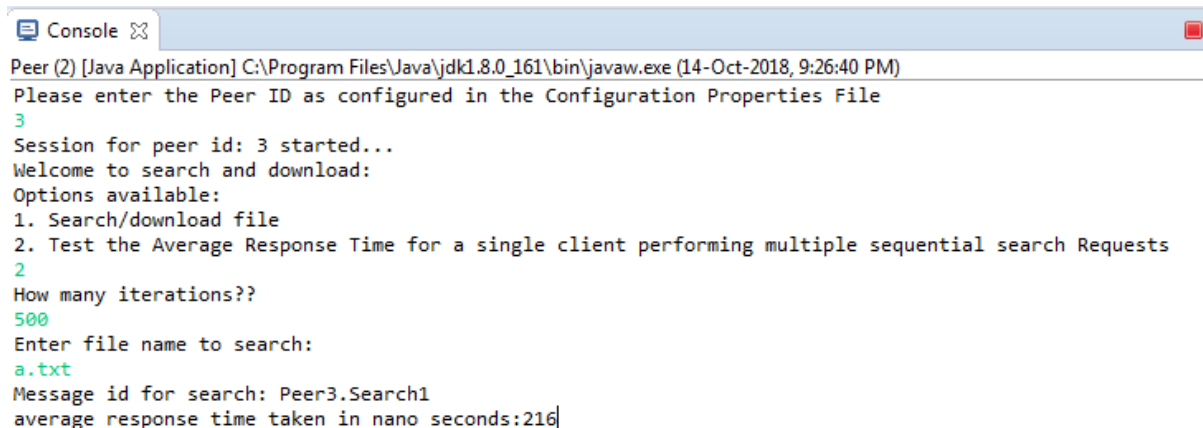
Figure 5: Screenshot of the leaf node demonstrating file download

To Test Average response time for multiple sequential search:

Precondition: The server should be running. And at least two clients should be registered.

In the Leaf node window,

- Press 2 to Start the test
- Enter the number of sequential search requests. Eg: 500
- Enter File name to be searched Eg: file2.txt
- The average response time for 500 sequential requests is displayed.



```
Peer (2) [Java Application] C:\Program Files\Java\jdk1.8.0_161\bin\javaw.exe (14-Oct-2018, 9:26:40 PM)
Please enter the Peer ID as configured in the Configuration Properties File
3
Session for peer id: 3 started...
Welcome to search and download:
Options available:
1. Search/download file
2. Test the Average Response Time for a single client performing multiple sequential search Requests
2
How many iterations??
500
Enter file name to search:
a.txt
Message id for search: Peer3.Search1
average response time taken in nano seconds:216
```

Figure 6: Screenshot of the leaf node demonstrating the calculation of average response time for a single client performing multiple sequential search requests