Programming Project – 4

CS5352 Advanced Operating Systems Design
Gnutella File Sharing System with push based approach
Output Screenshots

<div align="right">Manisha Siddartha Nalla</div>

Test Case-1: Setting config files
Mesh Topology(mesh.properties):

Star Topology(star.properties):

Result: pass

Test Case-2: Search operation:

A peer searches for files by issuing a query. The query is sent to all neighbors. Each neighbor looks up the specified file using a local index and responds with a hit query message in the event of a hit. The hit query message is propagated back to the original sender by following the reverse path of the query

Test result: pass

Test Case-3: Searching in the first level of neighbours

```
disci.hpcc.ttu.edu - PuTTY                                    —    □    ×
 2. exit
1
enter the file to search
file1.txt

 Neighbouring peers : peertwo peersix
 check in the port peertwo
File not found in peertwo

 check in the port peersix
File not found in peersix

 Neighbouring peers : peerone peerthree peerfive
 check in the port peerone
File hit in peerone
node already visited

 check in the port peerfive
Timer expired!node unreachable
(messageID ,[list of visited peers])
( 98,[peerthree, peertwo, peersix, peerone] )

 List of peers in which file is found  ( peerone ) peers
Enter the peer you want to get the file from :
```



```
disci.hpcc.ttu.edu - PuTTY                                    —    □    ×
 2. exit
1
enter the file to search
file1.txt

 Neighbouring peers : peertwo peersix
 check in the port peertwo
File not found in peertwo

 check in the port peersix
File not found in peersix

 Neighbouring peers : peerone peerthree peerfive
 check in the port peerone
File hit in peerone
node already visited

 check in the port peerfive
Timer expired!node unreachable
(messageID ,[list of visited peers])
( 98,[peerthree, peertwo, peersix, peerone] )

 List of peers in which file is found  ( peerone ) peers
Enter the peer you want to get the file from :
```

File is not found in the first level of neighbours

Test Result: pass

Testcase -4: Searching in next level of neighbours

Does not check again in already checked neighbours

```
disci.hpcc.ttu.edu - PuTTY                                    —    □    ×
 2. exit
1
enter the file to search
file1.txt

 Neighbouring peers : peertwo peersix
 check in the port peertwo
File not found in peertwo

 check in the port peersix
File not found in peersix

 Neighbouring peers :  peerone peerthree peerfive
 check in the port peerone
File hit in peerone
node already visited

 check in the port peerfive
Timer expired!node unreachable
(messageID ,[list of visited peers])
( 98,[peerthree, peertwo, peersix, peerone] )

 List of peers in which file is found  ( peerone ) peers
Enter the peer you want to get the file from :
```

Searching in the next level of neighbours

File hit in peer one

Test result: Pass

TestCase-5: Time to live value(TTL)


```
disci.hpcc.ttu.edu - PuTTY                                    —    □    ×
 2. exit
1
enter the file to search
file1.txt

 Neighbouring peers : peertwo peersix
 check in the port peertwo
File not found in peertwo

 check in the port peersix
File not found in peersix

 Neighbouring peers : peerone peerthree peerfive
 check in the port peerone
File hit in peerone
node already visited

 check in the port peerfive
Timer expired!node unreachable
(messageID ,[list of visited peers])
( 98,[peerthree, peertwo, peersix, peerone] )

 List of peers in which file is found  ( peerone ) peers
Enter the peer you want to get the file from :
```
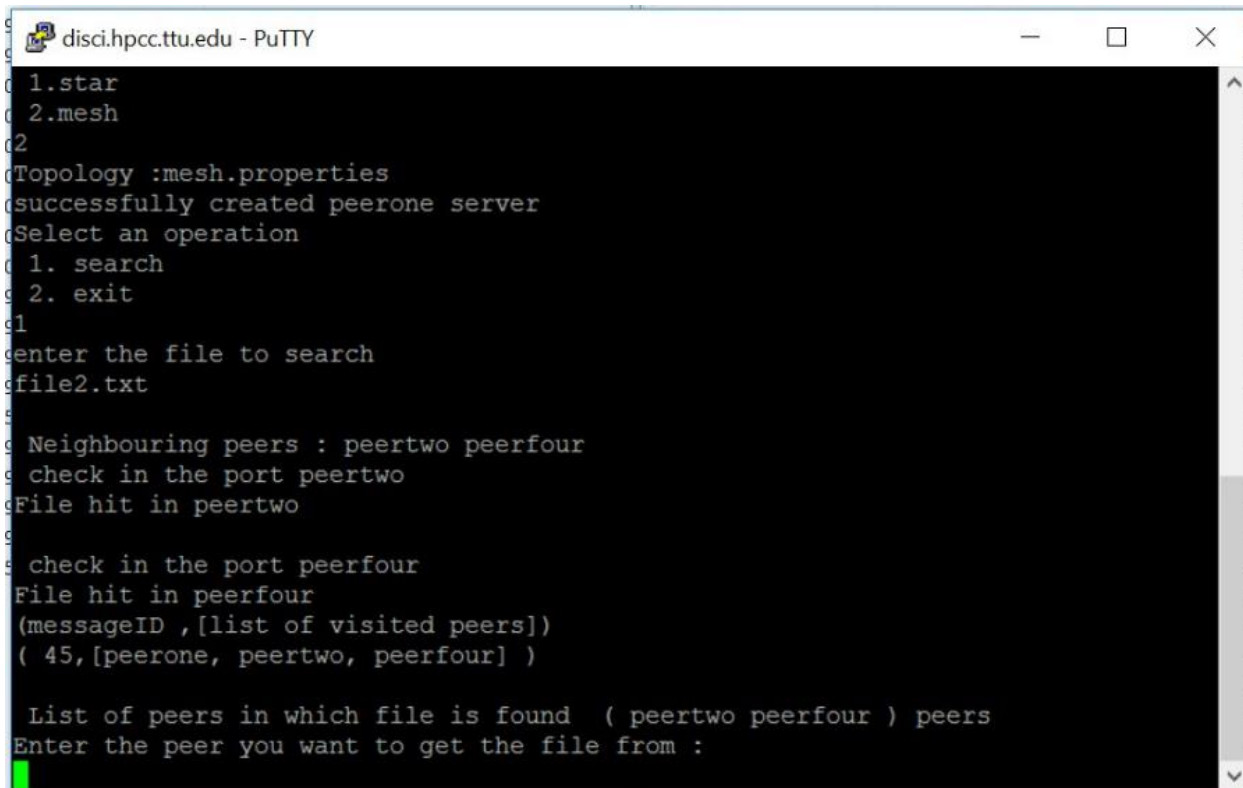
Time to live value decrements for each and every search operation and expires when it reaches 0. The search operation stops when it is expired. Hence it avoids searching infinitely.

TestCase-6: Searching in all its neighbors irrespective of hit or miss

```
disci.hpcc.ttu.edu - PuTTY                                    —    □    ✕
 1.star                                                                   ^
 2.mesh
2
Topology :mesh.properties
successfully created peerone server
Select an operation
 1. search
 2. exit
1
enter the file to search
file2.txt

 Neighbouring peers : peertwo peerfour
 check in the port peertwo
File hit in peertwo

 check in the port peerfour
File hit in peerfour
 (messageID ,[list of visited peers])
 ( 45,[peerone, peertwo, peerfour] )

 List of peers in which file is found  ( peertwo peerfour ) peers
Enter the peer you want to get the file from :
                                                                          ⌄
```
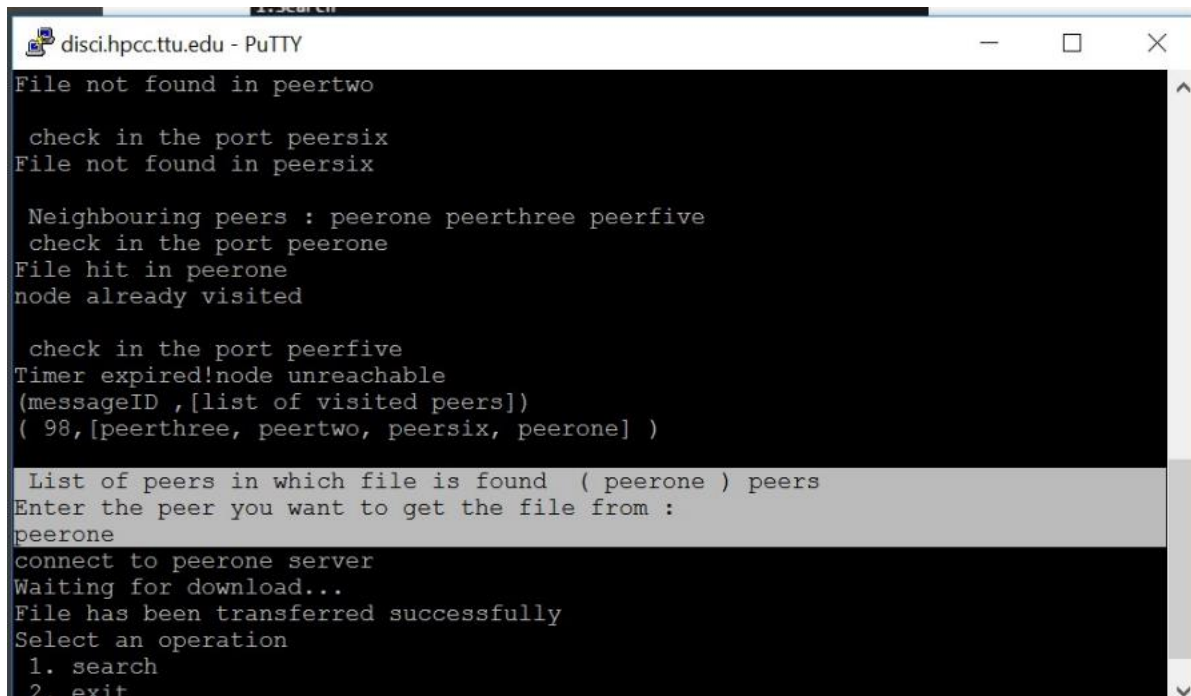
It checks in peerfour though it is a hit in peertwo

TestResult: pass

TestCase-7: Connecting to a peer from the list of peers found.


```
disci.hpcc.ttu.edu - PuTTY                                    —    □    ✕

File not found in peertwo

 check in the port peersix
File not found in peersix

 Neighbouring peers : peerone peerthree peerfive
 check in the port peerone
File hit in peerone
node already visited

 check in the port peerfive
Timer expired!node unreachable
(messageID ,[list of visited peers])
( 98,[peerthree, peertwo, peersix, peerone] )

 List of peers in which file is found  ( peerone ) peers
Enter the peer you want to get the file from :
peerone
connect to peerone server
Waiting for download...
File has been transferred successfully
Select an operation
 1. search
 2. exit
```
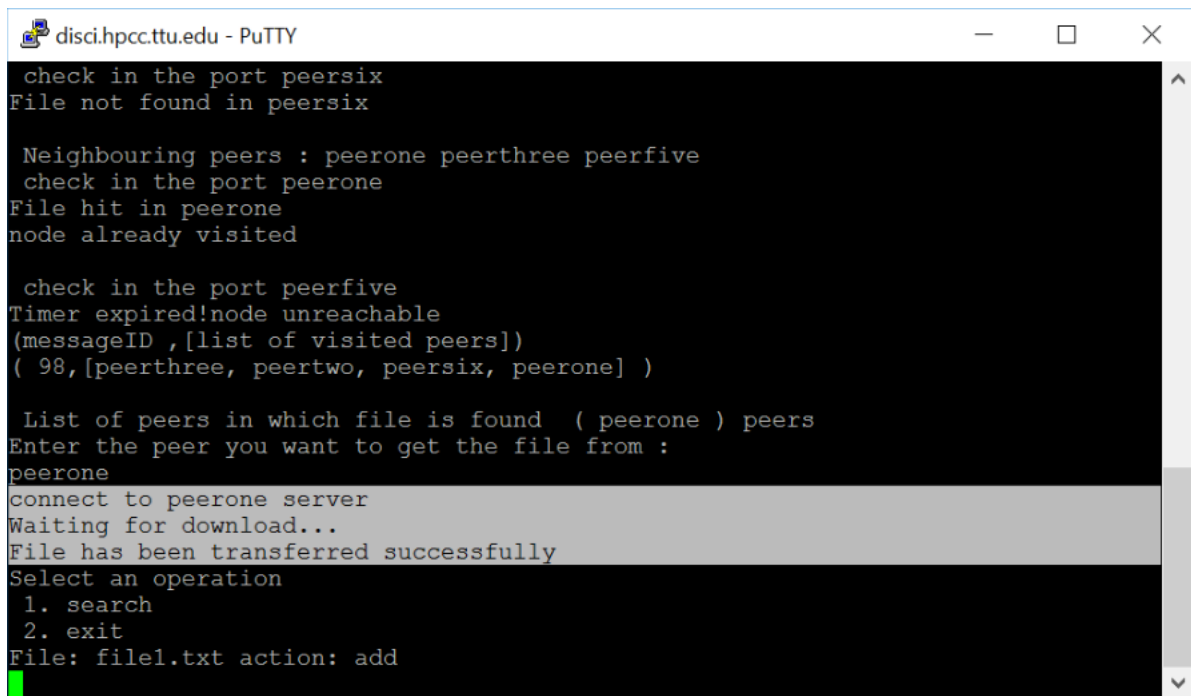
Test result: Pass

TestCase-8: Downloading the file


```
disci.hpcc.ttu.edu - PuTTY                                    —    □    ✕

 check in the port peersix
File not found in peersix

 Neighbouring peers : peerone peerthree peerfive
 check in the port peerone
File hit in peerone
node already visited

 check in the port peerfive
Timer expired!node unreachable
(messageID ,[list of visited peers])
( 98,[peerthree, peertwo, peersix, peerone] )

 List of peers in which file is found  ( peerone ) peers
Enter the peer you want to get the file from :
peerone
connect to peerone server
Waiting for download...
File has been transferred successfully
Select an operation
 1. search
 2. exit
File: file1.txt action: add
```
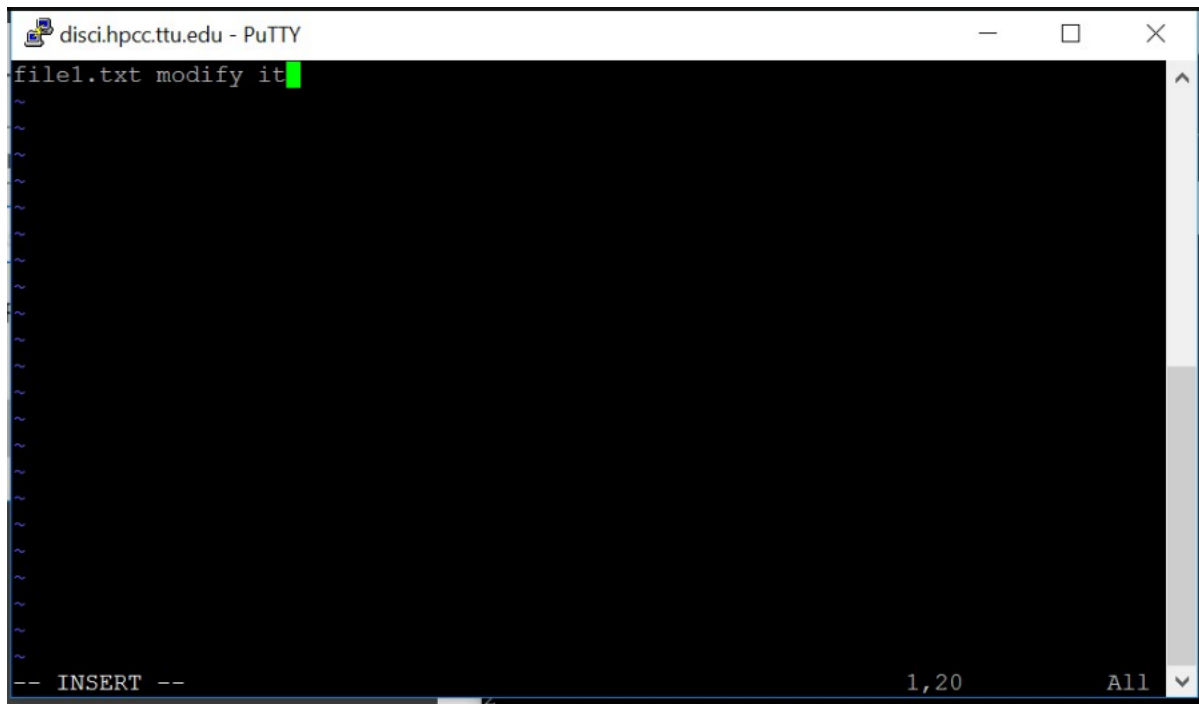
File has been downloaded successfully

TestResult:Pass

Test Case -9: push based approach

Monitors for changes in file and reflects them in cached files

Modified in master file of peer_7

Reflected in cached file of peer_3

TestResult: pass