

CSC/ECE 573- INTERNET **PROTOCOLS**

PROJECT 1 REPORT

SUNNY VERMA-200207927

sverma4@ncsu.edu

APARAJITA BAGCHI-200203869

adbagchi@ncsu.edu

TABLE OF CONTENTS

Sr. No	Title	Page No.
1.	Introduction	
2.	Message Formats	
3.	Download Time Curves	
4.	Discussion and Conclusions	
5.		
6.		
7.		

INTRODUCTION

When two or more hosts connect and share resources without the intervention of a separate server, we say that it is a peer to peer network.

The goal of this project is to implement a P2P-DI system that helps peers download RFCs that they may not have locally. The system consists of a RS server and 6 peers. Each peer has its own client and server. The RS server helps to register peers and provides a list of active peers when it is asked for a RFC. The querying peer then asks the active peers for the data and downloads it from the server of the peer who has it.

Note: Please follow the readme file to compile the code.

MESSAGE FORMATS

Peer to RS

Whenever a user tries to connect for the first time. The following messages will be displayed :

“What you want to do today “?

Enter 1 for Registering your peer:

Enter 2 for Leaving the P2P-DI system:

Enter 3 for Just PQuery from RS Server:

The user can now select the options from above.

First of all, The Peer will register itself with the registration server and choose the option as 1.

While registering, hostname and Port number are taken from the Peers and the same is stored in the database structure (Lists in this case) where the information of each peers is stored. RS server returns a cookie to the registered peer.

Second, if the peer leaves the P2P system, it will notify the Registration server that I'm leaving the system and choose the option as 2. The Registration server will update the list and make this peer inactive.

Third, when a peer wants to get the list of all the active peers, it will choose the option 3 from the list . The Registration server will fetch the data from the database (list in this case) where all the information about the active peers is stored and returns the list of all the active peers to this peer.

For example:

```
[{'peer_server_portnumber': 9004, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9006, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9001, 'hostname': '127.0.0.1'},
```

```
{'peer_server_portnumber': 9003, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9002, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9005, 'hostname': '127.0.0.1'}]
```

Peer to Peer

Enter 4 for Just RFCQuery from Active Peers:

The querying peer asks another peer for RFCQuery asking for the RFC index of the peer.

The function returns the list of all the RFC indexes of all the active peers.

For example:

```
Finally RFC LIST : [<peers.common.RFCs instance at 0x0000000002F9A208>,
<peers.common.RFCs instance at 0x0000000002F9A248>, <peers.common.RFCs instance at
0x0000000002F9A2C8>, <peers.common.RFCs instance at 0x0000000002F9A348>,
<peers.common.RFCs instance at 0x0000000002F9A3C8>, <peers.common.RFCs instance at
0x0000000002F9A448>, <peers.common.RFCs instance at 0x0000000002F9A4C8>,
<peers.common.RFCs instance at 0x0000000002F9A548>, <peers.common.RFCs instance at
0x0000000002F9A5C8>, <peers.common.RFCs instance at 0x0000000002F9A648>,
<peers.common.RFCs instance at 0x0000000002F9A6C8>, <peers.common.RFCs instance at
0x0000000002F9A748>, <peers.common.RFCs instance at 0x0000000002F9A7C8>,
<peers.common.RFCs instance at 0x0000000002F9A848>, <peers.common.RFCs instance at
0x0000000002F9A8C8>, <peers.common.RFCs instance at 0x0000000002F9A948>,
<peers.common.RFCs instance at 0x0000000002F9A9C8>, <peers.common.RFCs instance at
0x0000000002F9AA48>, <peers.common.RFCs instance at 0x0000000002F9AAC8>,
<peers.common.RFCs instance at 0x0000000002F9AB48>, <peers.common.RFCs instance at
0x0000000002F9ABC8>, <peers.common.RFCs instance at 0x0000000002F9AC48>,
<peers.common.RFCs instance at 0x0000000002F9ACC8>, <peers.common.RFCs instance at
0x0000000002F9AD48>, <peers.common.RFCs instance at 0x0000000002F9ADC8>,
<peers.common.RFCs instance at 0x0000000002F9AE48>, <peers.common.RFCs instance at
0x0000000002F9AEC8>, <peers.common.RFCs instance at 0x0000000002F9AF48>,
<peers.common.RFCs instance at 0x0000000002F9AFC8>, <peers.common.RFCs instance at
```

0x0000000002F99088>, <peers.common.RFCs instance at 0x0000000002F99108>, <peers.common.RFCs instance at 0x0000000002F99188>, <peers.common.RFCs instance at 0x0000000002F99208>, <peers.common.RFCs instance at 0x0000000002F99288>, <peers.common.RFCs instance at 0x0000000002F99308>, <peers.common.RFCs instance at 0x0000000002F99388>, <peers.common.RFCs instance at 0x0000000002F99408>, <peers.common.RFCs instance at 0x0000000002F99488>, <peers.common.RFCs instance at 0x0000000002F99508>, <peers.common.RFCs instance at 0x0000000002F99588>, <peers.common.RFCs instance at 0x0000000002F99608>, <peers.common.RFCs instance at 0x0000000002F99688>, <peers.common.RFCs instance at 0x0000000002F99708>, <peers.common.RFCs instance at 0x0000000002F99788>, <peers.common.RFCs instance at 0x0000000002F99808>, <peers.common.RFCs instance at 0x0000000002F99888>, <peers.common.RFCs instance at 0x0000000002F99908>, <peers.common.RFCs instance at 0x0000000002F99988>, <peers.common.RFCs instance at 0x0000000002F99A08>, <peers.common.RFCs instance at 0x0000000002F99A88>, <peers.common.RFCs instance at 0x0000000002F99B08>, <peers.common.RFCs instance at 0x0000000002F99B88>, <peers.common.RFCs instance at 0x0000000002F99C08>, <peers.common.RFCs instance at 0x0000000002F99C88>, <peers.common.RFCs instance at 0x0000000002F99D08>, <peers.common.RFCs instance at 0x0000000002F99D88>, <peers.common.RFCs instance at 0x0000000002F99E08>, <peers.common.RFCs instance at 0x0000000002F99E88>, <peers.common.RFCs instance at 0x0000000002F99F08>, <peers.common.RFCs instance at 0x0000000002F99F88>]

Finally LOCALRFC LIST : []

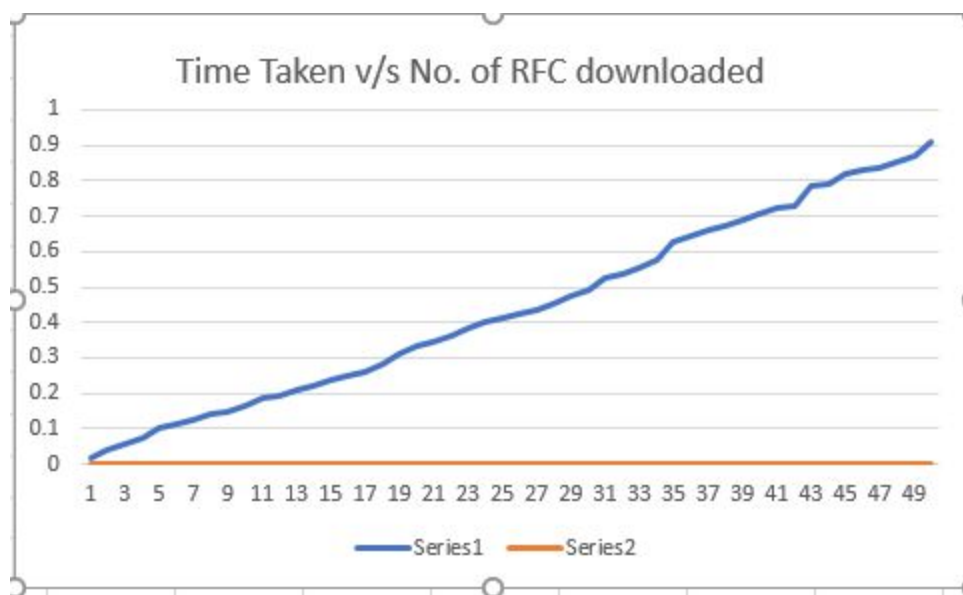
Enter 5 for All downloading the RFCs:

The peers can download the RFC's whichever is needed from the active peers.

DOWNLOAD TIME CURVES

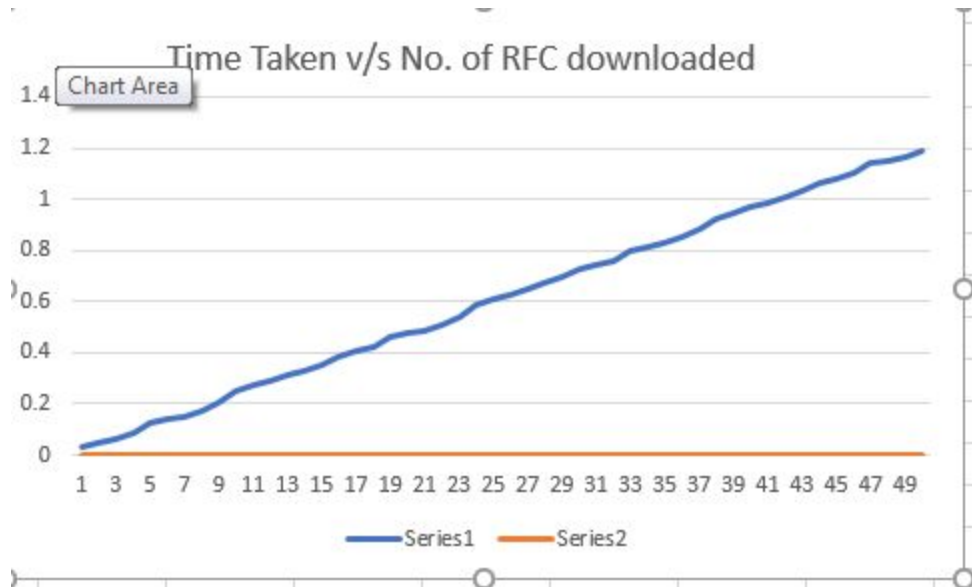
Task 1

In this task, Peer 1 has all 60 RFCs while Peer 2 to Peer 6 have none. They send RFCQuery to Peer 1 asking to download the files. Each peer downloads 50 files from peer 1 and records the time taken. The plot of the time taken vs the number of RFCs is as follows:



We got similar results in all other peers.

Task 2:



DISCUSSIONS AND CONCLUSIONS

Centralized Distribution Architecture: We consider the *best case* where all clients do not flood the server at once. This does not overload the server and hence it does not take much time to download the files.

The *worst case* happens when all peers attempt to access the same time. This overloads the server which then leads to congestion. Obviously it takes more time in this situation to download the files.

Peer to peer Architecture: Less time is taken to download the file than the centralized server architecture in the *best case* when all files are equally distributed among the peers.

In the *worst case* the system works the same as centralized server architecture as the first peer contains all the files and rest of the peers are trying to access them all at once.

To conclude, peer to peer transfer proves to be more efficient than centralized server architecture.

```
User choose 1
Now we will connect to RS Server and register ourself.
Connected
Server sent "Status: 200 OK, cookie: 1001 "
Client got cookie : 1001 from the server
```

For PQuery:

What you want to do today?

Enter 1 for Registering your peer:

Enter 2 for Leaving the P2P-DI system:

Enter 3 for Just PQuery from RS Server:

Enter 4 for Just RFCQuery from Active Peers:

Enter 5 for All downloading the RFCs:

3

User choose 3

Peer want to do PQuery

Now client will connect to server and ask for Active Peer list

```
[{'peer_server_portnumber': 9004, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9006,
'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9001, 'hostname': '127.0.0.1'},
{'peer_server_portnumber': 9003, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9002,
'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9005, 'hostname': '127.0.0.1'}]
```

```

What you want to do today?
Enter 1 for Registering your peer:
Enter 2 for Leaving the P2P-DI system:
Enter 3 for Just PQuery from RS Server:
Enter 4 for Just RFCQuery from Active Peers:
Enter 5 for All downloading the RFCs:
3
User choose 3
Peer want to do PQuery

Now client will connect to server and ask for Active Peer list
[{'peer_server_portnumber': 9004, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9006, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9001, 'hostname': '127.0.0.1'}, {'peer_server_portnumber': 9003, 'hos

```

Query Active Peers:

What you want to do today?

Enter 1 for Registering your peer:

Enter 2 for Leaving the P2P-DI system:

Enter 3 for Just PQuery from RS Server:

Enter 4 for Just RFCQuery from Active Peers:

Enter 5 for All downloading the RFCs:

4

User choose 4

Initially : [] []

Finally RFC LIST : [<peers.common.RFCs instance at 0x000000002F9A208>, <peers.common.RFCs instance at 0x000000002F9A248>, <peers.common.RFCs instance at 0x000000002F9A2C8>, <peers.common.RFCs instance at 0x000000002F9A348>, <peers.common.RFCs instance at 0x000000002F9A3C8>, <peers.common.RFCs instance at 0x000000002F9A448>, <peers.common.RFCs instance at 0x000000002F9A4C8>, <peers.common.RFCs instance at 0x000000002F9A548>, <peers.common.RFCs instance at 0x000000002F9A5C8>, <peers.common.RFCs instance at 0x000000002F9A6C8>, <peers.common.RFCs instance at 0x000000002F9A748>, <peers.common.RFCs instance at 0x000000002F9A7C8>, <peers.common.RFCs instance at 0x000000002F9A848>, <peers.common.RFCs instance at 0x000000002F9A8C8>, <peers.common.RFCs instance at 0x000000002F9A948>, <peers.common.RFCs instance at 0x000000002F9A9C8>, <peers.common.RFCs instance at 0x000000002F9AA48>, <peers.common.RFCs instance at 0x000000002F9AAC8>, <peers.common.RFCs instance at 0x000000002F9AB48>, <peers.common.RFCs instance at 0x000000002F9ABC8>, <peers.common.RFCs instance at 0x000000002F9AC48>, <peers.common.RFCs instance at 0x000000002F9ACC8>, <peers.common.RFCs instance at

0x0000000002F9AD48>, <peers.common.RFCs instance at 0x0000000002F9ADC8>,
<peers.common.RFCs instance at 0x0000000002F9AE48>, <peers.common.RFCs instance at
0x0000000002F9AEC8>, <peers.common.RFCs instance at 0x0000000002F9AF48>,
<peers.common.RFCs instance at 0x0000000002F9AFC8>, <peers.common.RFCs instance at
0x0000000002F99088>, <peers.common.RFCs instance at 0x0000000002F99108>,
<peers.common.RFCs instance at 0x0000000002F99188>, <peers.common.RFCs instance at
0x0000000002F99208>, <peers.common.RFCs instance at 0x0000000002F99288>,
<peers.common.RFCs instance at 0x0000000002F99308>, <peers.common.RFCs instance at
0x0000000002F99388>, <peers.common.RFCs instance at 0x0000000002F99408>,
<peers.common.RFCs instance at 0x0000000002F99488>, <peers.common.RFCs instance at
0x0000000002F99508>, <peers.common.RFCs instance at 0x0000000002F99588>,
<peers.common.RFCs instance at 0x0000000002F99608>, <peers.common.RFCs instance at
0x0000000002F99688>, <peers.common.RFCs instance at 0x0000000002F99708>,
<peers.common.RFCs instance at 0x0000000002F99788>, <peers.common.RFCs instance at
0x0000000002F99808>, <peers.common.RFCs instance at 0x0000000002F99888>,
<peers.common.RFCs instance at 0x0000000002F99908>, <peers.common.RFCs instance at
0x0000000002F99988>, <peers.common.RFCs instance at 0x0000000002F99A08>,
<peers.common.RFCs instance at 0x0000000002F99A88>, <peers.common.RFCs instance at
0x0000000002F99B08>, <peers.common.RFCs instance at 0x0000000002F99B88>,
<peers.common.RFCs instance at 0x0000000002F99C08>, <peers.common.RFCs instance at
0x0000000002F99C88>, <peers.common.RFCs instance at 0x0000000002F99D08>,
<peers.common.RFCs instance at 0x0000000002F99D88>, <peers.common.RFCs instance at
0x0000000002F99E08>, <peers.common.RFCs instance at 0x0000000002F99E88>,
<peers.common.RFCs instance at 0x0000000002F99F08>, <peers.common.RFCs instance at
0x0000000002F99F88>]

Finally LOCALRFC LIST : []