Project Submitted by:

Atul Sharma and Rashi Agarwal

1. File name and their uses
2. RS\_Server.py
3. Client-A.py
4. Client-A-RFC\_Server.py
5. Client-B.py
6. Client-B-RFC\_Server.py

RS\_server.py :

* RS Server will accept the registration request for all the peers in the network on well-known port 65423.
* Peer will send their listening port to the RS server which is hardcoded.
* RS Server will maintain the record of the peers with their IP Address, hostname, port number, listening port, Time to live, Registered count and time of registration.
* Cookie will be assigned by the RS Server to the peer.
* RS Server will keep all these records in dictionary.
* RS Server will share peer list to the peers on request basis.

ClientA.py:

* Client will open the connection to the RS Server with the port 65423
* Client will have the menu options where user can pass the inputs for Registration, Leave, PQuery from 1 to 3 respectively

1. Registration
2. Leave
3. PQuery – In this there is sub option –

(1). RFC Query

Client-A-RFC\_Server.py

* RFC Server is accepting all other peers RFC Client request and send the RFC document.
* RFC server listening port is set to 65425
* Peer A already has that RFC.
* In our case ‘rfc123’ is the sample rfc which is being requested by the peer B.
* Rfc is being stored in the program directory

Client-B:

* Client B is registering to RS Server with port 65423 and listening port 65425
* Client B is requesting for RFC- rfc123 from Peer A and downloading.

Assumptions:

* RS Server is assuming Peer is always connected until forcefully quit the program.
* Peer A has 1 file ‘rfc123.txt’ and Peer B downloading the rfc123.txt
* Cookie is being sent to Peer via Peer list which contains all the peer information
* Both RFC Servers are listening to same port