Peer-to-Peer with Centralized Index (P2P-CI) System for Downloading RFCs

-AEKLASP
-HBALAJI

Design

The system is designed as per the specifications mentioned in the Project document (proj1.pdf). Some of the design questions that were unanswered in the specification for which we made appropriate assumptions are the following

- Data sent between the server and client or client or client is serialized before sending and the serialization technique used is CPickle (inbuilt functionality in Python)
- To provide concurrency at both Server and Client, we are implementing their functionalities as threads, we are also implementing thread locking mechanism to maintain data consistency
- At Client
    - Two threads are implemented, one to server GET requests from other clients and another to provide console interface to User
- At Server
    - One thread is created for each client that connects with the server.
    - The details of the peers are stored as as list - [[peer, uport], [peer, uport]]
    - The details of RFC are stored as a list - [[RFC no, RFC title, hostName],[RFC no, RFC title, hostName].....]
- The specification is not clear about how a client exist the system, so we have assumed that a client can just close the connections and server will catch this as an exception and do the required client deregistration process (remove from database)

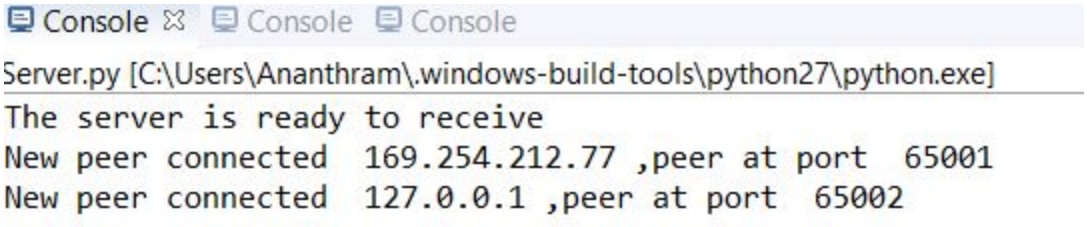
Basic System Testing:

System under test:

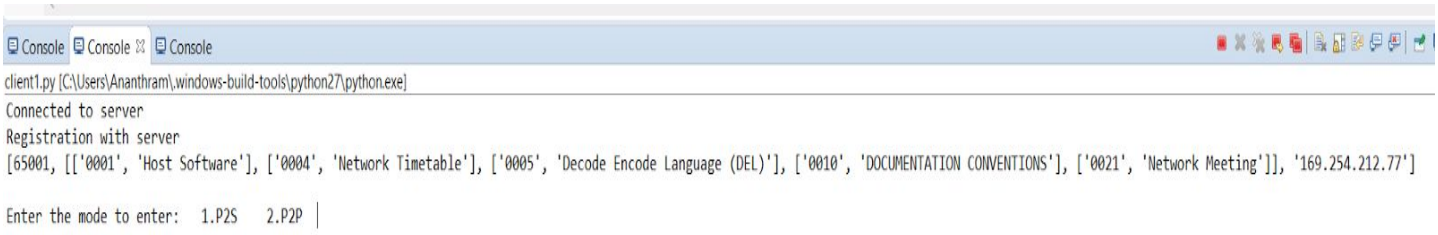
- Server.py running (listening at port 7734, IP address as host machine)
- client1.py running with upload port 65001 and IP as IP of host machine. Client1's RFC directory is rfc folder
- client2.py running with upload port 65002 and IP address as 127.0.0.1. Client2's RFC directory is rfc2 folder

Test case 1: Both clients connect to server and register by sending their local RFCs .

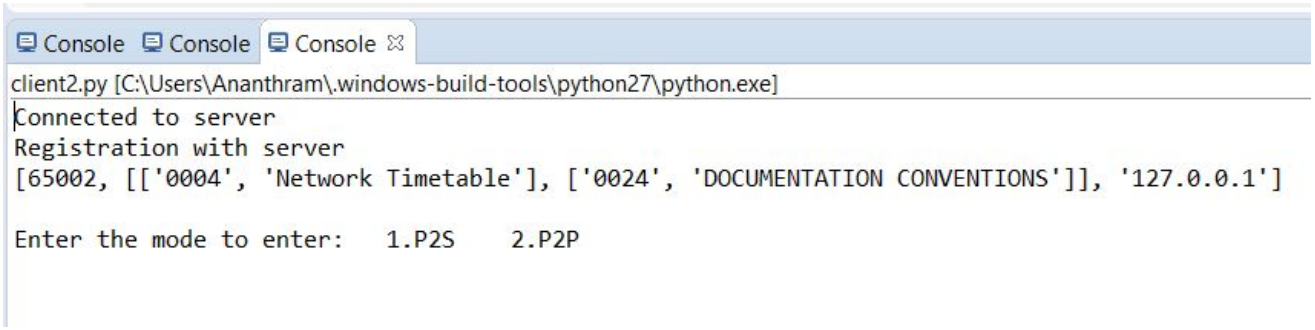
At server:



At Client1



At Client 2



Testcase 2 : Peer to server communication , LIST command

At Client 1

```
Enter the mode to enter:   1.P2S    2.P2P  P2S
Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
LIST
Request to server.....
LIST ALL P2P-CI/1.0
Host: 169.254.212.77
Port: 65001


Response from server....
P2P-CI/1.0 200 OK
RFC 0001 Host Software 169.254.212.77 65001
RFC 0004 Network Timetable 169.254.212.77 65001
RFC 0005 Decode Encode Language (DEL) 169.254.212.77 65001
RFC 0010 DOCUMENTATION CONVENTIONS 169.254.212.77 65001
RFC 0021 Network Meeting 169.254.212.77 65001
RFC 0004 Network Timetable 127.0.0.1 65002
RFC 0024 DOCUMENTATION CONVENTIONS 127.0.0.1 65002

Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
```

TESTCASE 3: Peer to server communication , LOOKUP command

At Client 1, Lookup for RFC 0004 (available) and RFC 0067 (not available)

```
Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
LOOKUP
Enter the RFC Number:0004
Enter the RFC Title:Network Timetable
Request to server.....
LOOKUP RFC 0004 P2P-CI/1.0
Host: 169.254.212.77
Port: 65001
Title: Network Timetable


Response from server....
P2P-CI/1.0 200 OK
RFC 0004 Network Timetable 169.254.212.77 65001
RFC 0004 Network Timetable 127.0.0.1 65002

Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
LOOKUP
Enter the RFC Number:0067
Enter the RFC Title:Proposed Change to Host/IMP Spec
Request to server.....
LOOKUP RFC 0067 P2P-CI/1.0
Host: 169.254.212.77
Port: 65001
Title: Proposed Change to Host/IMP Spec


Response from server....
P2P-CI/1.0 404 Not Found

Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
```

TESTCASE 4: Peer to server communication , ADD command

Adding RFC 97 from client 1 and checking the updated RFC List at server by using the LIST command from Client 1

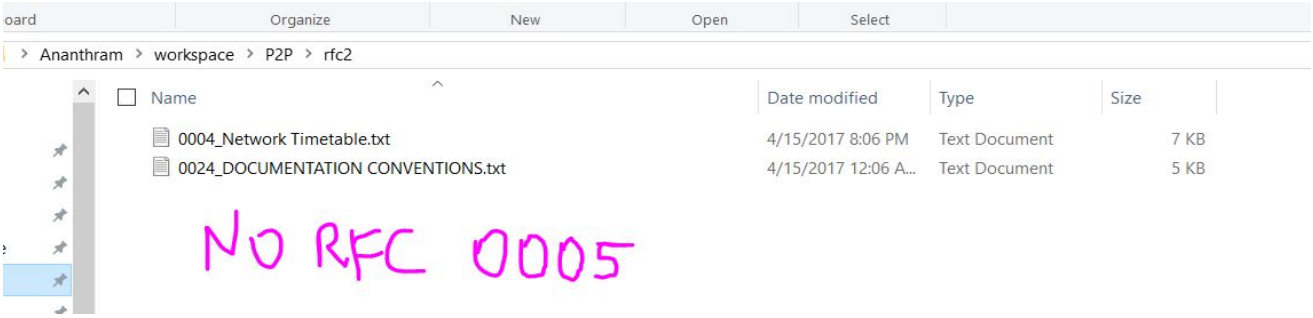
```
Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
ADD
Enter the RFC Number:0097
Enter the RFC Title: FIRST CUT AT A PROPOSED TELNET PROTOCOL
Request to server.....
ADD RFC 0097 P2P-CI/1.0
Host:  169.254.212.77
Port:  65001
Title:  FIRST CUT AT A PROPOSED TELNET PROTOCOL


Response from server....
P2P-CI/1.0 200 OK
RFC 0097 FIRST CUT AT A PROPOSED TELNET PROTOCOL 169.254.212.77 65001
Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
LIST
Request to server.....
LIST ALL P2P-CI/1.0
Host: 169.254.212.77
Port: 65001
```

```
Response from server....
P2P-CI/1.0 200 OK
RFC 0001 Host Software 169.254.212.77 65001
RFC 0004 Network Timetable 169.254.212.77 65001
RFC 0005 Decode Encode Language (DEL) 169.254.212.77 65001
RFC 0010 DOCUMENTATION CONVENTIONS 169.254.212.77 65001
RFC 0021 Network Meeting 169.254.212.77 65001
RFC 0004 Network Timetable 127.0.0.1 65002
RFC 0024 DOCUMENTATION CONVENTIONS 127.0.0.1 65002
RFC 0097 FIRST CUT AT A PROPOSED TELNET PROTOCOL 169.254.212.77 65001

Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
```

TESTCASE 5: Peer to Peer communication , GET command

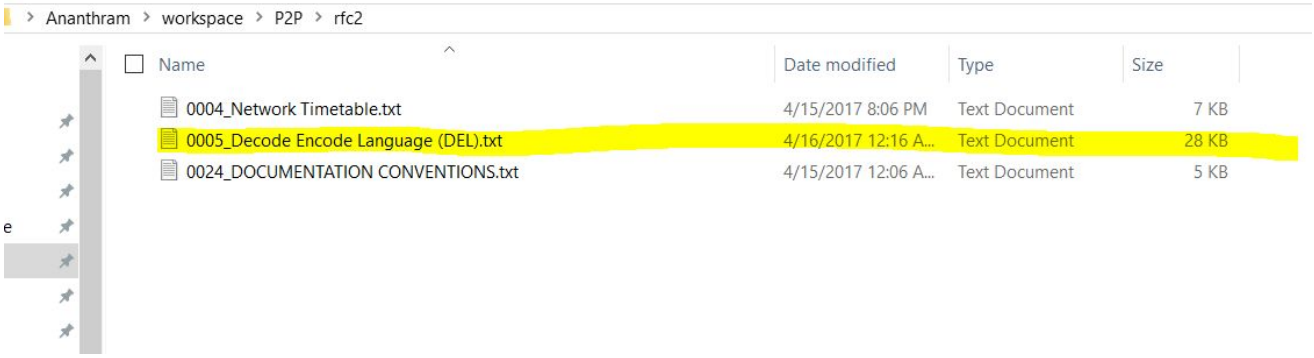
Getting RFC 5 from Client1 to Client2

At Client 2 directory:

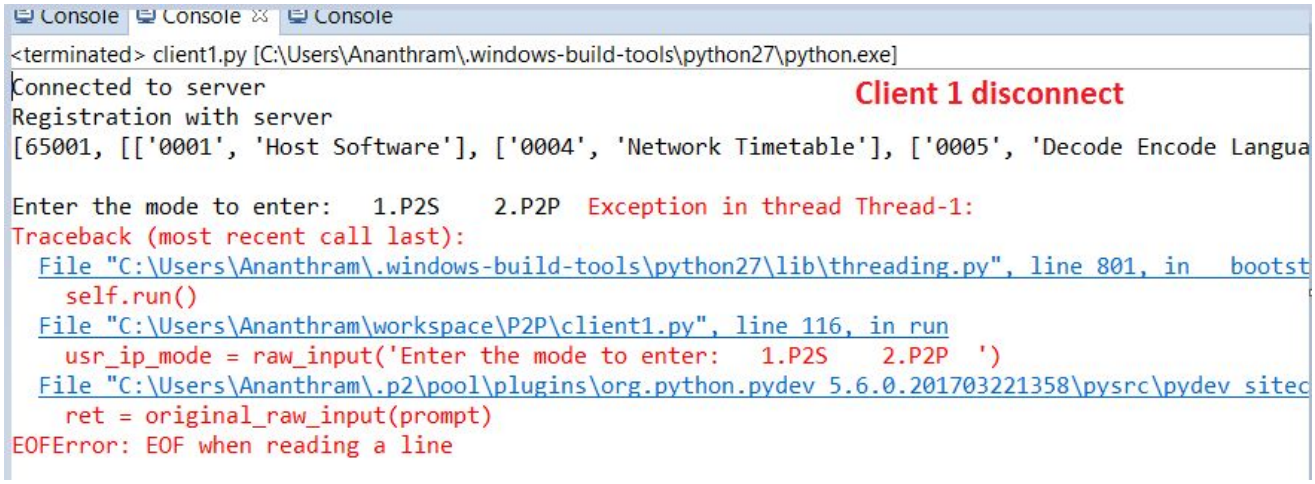


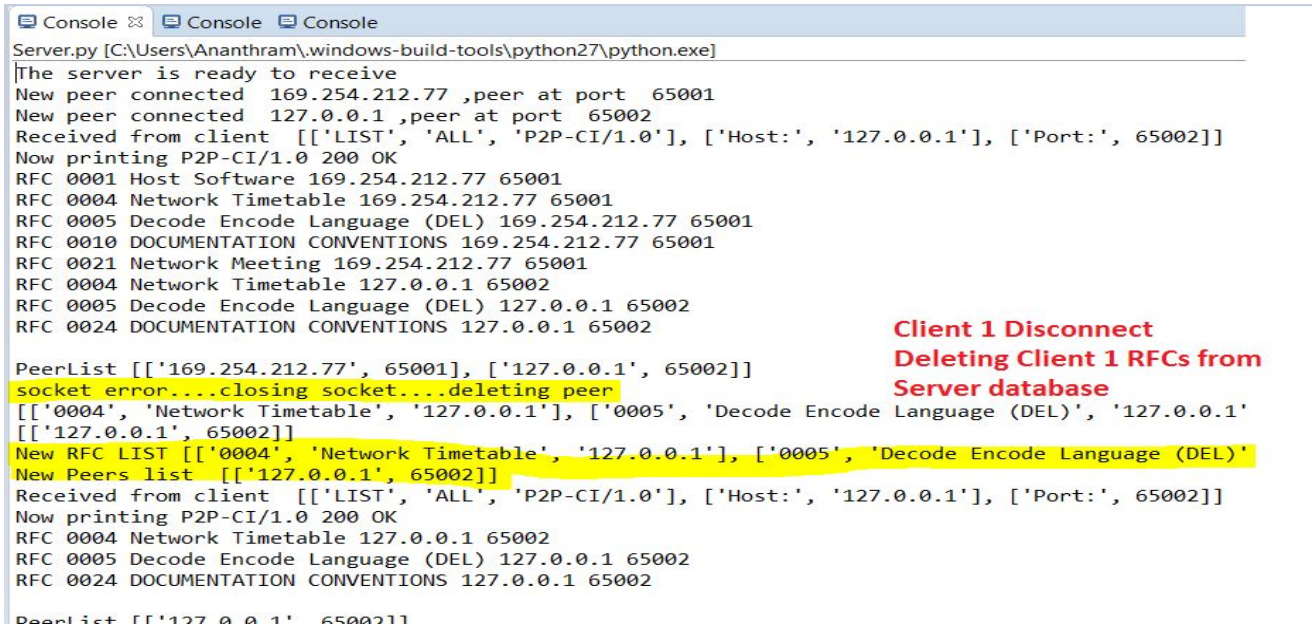

```
client2.py [C:\Users\Ananthram\.windows-build-tools\python27\python.exe]
3.LOOKUP
4.EXIT
EXIT
Enter the mode to enter:   1.P2S    2.P2P  P2P
Enter your input
1.GET
2.EXIT
GET
Enter Client IP: 169.254.212.77
Enter Client port: 65001
Enter the RFC Number: 0005
Enter the RFC Name: Decode Encode Language (DEL)
Request to server.....
GET RFC 0005 P2P-CI/1.0
Host: 127.0.0.1
OS: Windows

Response from server....
P2P-CI/1.0 200 OK
Date: Sun, 16 Apr 2017 00:16:54 Eastern Daylight Time
OS: Windows
Last-Modified: Sat Apr 16 14:02:02 2016
Content-Length: 27583
Content-Type: text/text

Network Working Group                                          4691
RFC-5                                                          Jeff Rulifson
                                                               June 2, 1969

                                    DEL

RFC Printed

:DEL, 02/06/69 1010:58   JFR    ;    .DSN=1; .LSP=0; ['=] AND NOT SP ; ['?];
```

After RFC 0005 Now available at Client2 directory



TESTCASE 6: Client Disconnect scenario, RFCs deleted from server RFC Index





Since Client 2 is still connected to the server, LIST at Client 2 does not list Client 1 RFCs now

```
Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
```

```
client2.py [C:\Users\Ananthram\.windows-build-tools\python27\python.exe]
4.EXIT
LIST
Request to server.....
LIST ALL P2P-CI/1.0
Host: 127.0.0.1
Port: 65002


Response from server....
P2P-CI/1.0 200 OK
RFC 0001 Host Software 169.254.212.77 65001
RFC 0004 Network Timetable 169.254.212.77 65001
RFC 0005 Decode Encode Language (DEL) 169.254.212.77 65001
RFC 0010 DOCUMENTATION CONVENTIONS 169.254.212.77 65001
RFC 0021 Network Meeting 169.254.212.77 65001
RFC 0004 Network Timetable 127.0.0.1 65002
RFC 0005 Decode Encode Language (DEL) 127.0.0.1 65002
RFC 0024 DOCUMENTATION CONVENTIONS 127.0.0.1 65002

Enter your input
1.ADD
2.LIST
3.LOOKUP
4.EXIT
LIST
Request to server.....
LIST ALL P2P-CI/1.0
Host: 127.0.0.1
Port: 65002


Response from server....
P2P-CI/1.0 200 OK
RFC 0004 Network Timetable 127.0.0.1 65002
RFC 0005 Decode Encode Language (DEL) 127.0.0.1 65002
RFC 0024 DOCUMENTATION CONVENTIONS 127.0.0.1 65002
```

We see that Client 1 RFC's are not listed when LIST command is used at Client 2