

# CSC 573 – Internet Protocols

## Project 1

**Archit Kwatra – 200316390**

Please note: The below instructions are for windows machine. Also, there are some libraries required which were pre-included with the python on windows. I have mentioned the one which I had to install. Please feel free to install other required libraries in case there is an error.

Requirements:

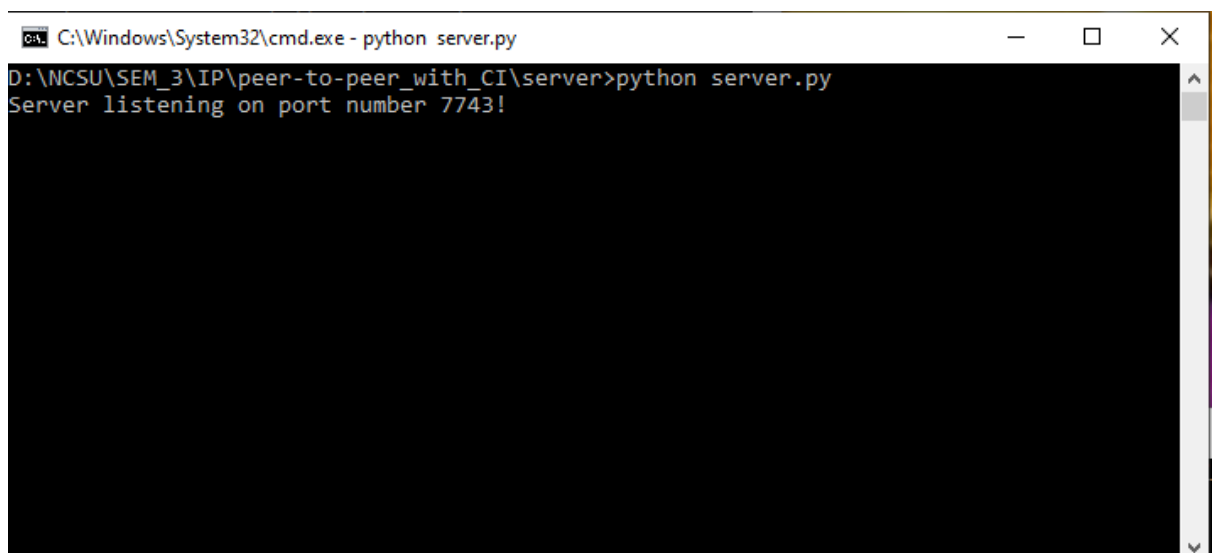
- Python 3.6 or above
- pytz library for time zone information (separately installed)

The project directory has 4 main folders:

- server – Folder for the server and its components
- client1 – First peer/client
- client2 – Second peer/client
- client3 – Third peer/client

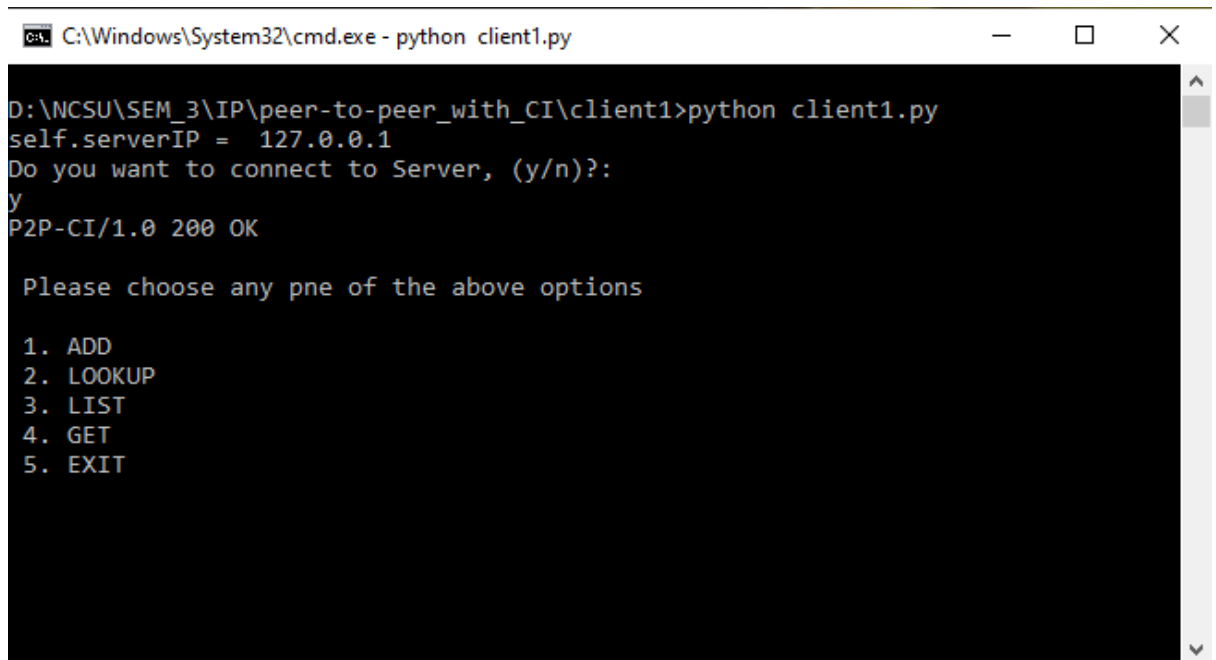
Steps for testing (run the first 4 commands on different terminal windows) -:

- 1) Go the server folder and run the command “python server.py”. You will see the below terminal state after running the above command.



```
C:\Windows\System32\cmd.exe - python server.py
D:\NCSU\SEM_3\IP\peer-to-peer_with_CI\server>python server.py
Server listening on port number 7743!
```

- 2) Go to client1 folder and run the command “python client1.py”, type y and press enter. You will see the below terminal state after running the above command.

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\System32\cmd.exe - python client1.py'. The command prompt shows the execution of 'python client1.py' in the directory 'D:\NCSU\SEM\_3\IP\peer-to-peer\_with\_CI\client1'. The script sets 'self.serverIP = 127.0.0.1' and asks 'Do you want to connect to Server, (y/n)?'. The user inputs 'y', and the script outputs 'P2P-CI/1.0 200 OK'. It then prompts 'Please choose any pne of the above options' and lists five options: 1. ADD, 2. LOOKUP, 3. LIST, 4. GET, and 5. EXIT.

```
C:\Windows\System32\cmd.exe - python client1.py

D:\NCSU\SEM_3\IP\peer-to-peer_with_CI\client1>python client1.py
self.serverIP = 127.0.0.1
Do you want to connect to Server, (y/n)?
y
P2P-CI/1.0 200 OK

Please choose any pne of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT
```

- 3) Go to client2 folder and run the command “python client2.py”, type y and press enter. The output will be very much same as the one in step 2
- 4) Go to client3 folder and run the command “python client3.py”, type y and press enter. The output will be very much same as the one in step 2 (This client testing is not necessary, and you can skip this if you want to. Two clients/peers are enough for testing purposes)
- 5) Now, press 1 in client1 and hit enter. It will ask you to enter the RFC number you wish to add. Once you type the RFC number and press enter, it will ask you to enter the RFC title. Type the required information and hit enter. The RFC will be added to the current RFC list. Below are the output snapshot examples for reference. For adding RFCs from other peers, you can follow the same steps from their respective terminals

```
Ca Select C:\Windows\System32\cmd.exe - python client1.py
D:\NCSU\SEM_3\IP\peer-to-peer_with_CI\client1>python client1.py
self.serverIP = 127.0.0.1
Do you want to connect to Server, (y/n)? :
y
P2P-CI/1.0 200 OK

Please choose any one of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT

1
Enter RFC number:
1
Enter RFC title:
rfc1
P2P-CI/1.0 200 OK
1 rfc1 Archit@928 29270

Please choose any one of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT
```

- 6) You can now check all the added RFCs by typing 3 and hitting enter.
- 7) For an RFC lookup, press 2 and hit enter. It will ask you the RFC number and display the RFC information if it is available with any other of the peers, otherwise it will display the required message. Please see the below snapshot for your reference.

```
C:\Windows\System32\cmd.exe - python client1.py

1
Enter RFC number:
10
Enter RFC title:
abc
P2P-CI/1.0 200 OK
10 abc Archit@928 29270

Please choose any pne of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT

3
P2P-CI/1.0 200 OK
10 abc Archit@928 29270
1 rfc1 Archit@928 29270

Please choose any pne of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT
```

- 8) To download an RFC from any peer, type 4 and hit enter. It will ask you the RFC number. If the RFC will be available with any of the peers, it will download it, otherwise, it will display an error. Once downloaded, it will add the RFC from that peer to the RFC list as well automatically. Please see the below snapshot for reference.

```
C:\Windows\System32\cmd.exe - python client3.py

D:\NCSU\SEM_3\IP\peer-to-peer_with_CI\client3>python client3
.py
self.serverIP = 127.0.0.1
Do you want to connect to Server, (y/n)? :
y
P2P-CI/1.0 200 OK

Please choose any one of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT

4
Enter RFC number:
10

Please choose any one of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT

P2P-CI/1.0 200 OK
10 abc Archit@928 29270
RFC downloaded
P2P-CI/1.0 200 OK
10 abc Archit@190 36222

Please choose any one of the above options

1. ADD
2. LOOKUP
3. LIST
4. GET
5. EXIT
```

- 9) Press 5 and hit enter to exit. All the information for the RFC will be automatically removed from the lists.

REF:

- 1) [https://en.wikipedia.org/wiki/Peer-to-peer\\_file\\_sharing](https://en.wikipedia.org/wiki/Peer-to-peer_file_sharing)
- 2) <http://www.cloudbus.org/papers/P2PbasedContentSharing.pdf>
- 3) <http://www.stackoverflow.com>

- 4) <https://medium.com/@amannagpal4/how-to-create-your-own-decentralized-file-sharing-service-using-python-2e00005bdc4a>