**CSC 573 – Internet Protocols**

**Project #1**

**Spring 2019**

**READ-ME/REPORT**

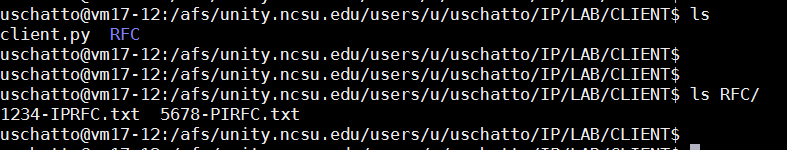
***NOTE: We have used Python 2.7.12 version. We have made use of os, sys, threading, socket, time, datetime, random and platform modules in our code.***

We have attached server.py and client.py files along with two sample RFCs for testing purposes. Please follow the below instructions before executing the scripts.

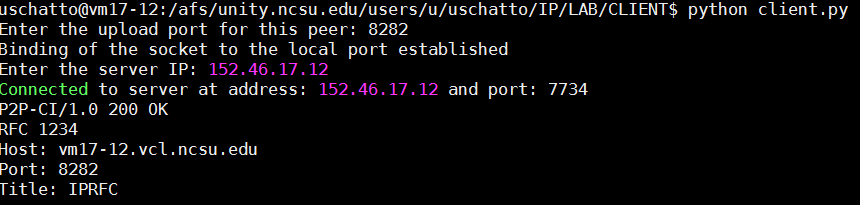
1. Run the server on a machine with the command – python server.py

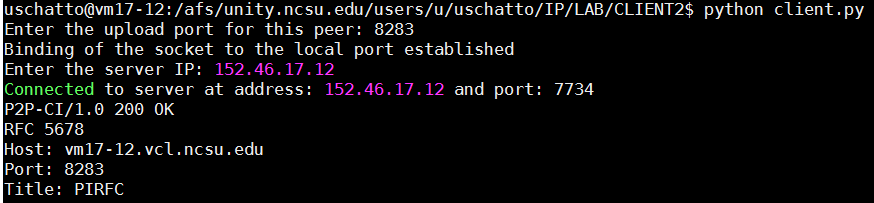


1. Before running the peer, ensure that the RFC folder is in place with the RFC files in that folder. The format of the RFC file should be <RFC\_NUMBER>-<RFC\_TITLE>.txt. For example- 5678-PIRFC.txt where 5678 is the RFC number and PIRFC is the RFC title. A sample file/directory structure format:

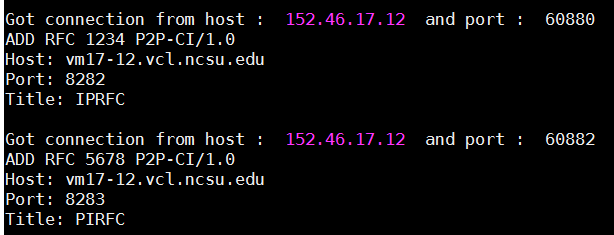


1. Run the client on another machine with the command - python client.py
   1. As soon as you run the script – A prompt will ask for the user to enter the upload port for that peer. It then asks for the server IP address. Ultimately it makes a connection to the server on the well-known port 7734 and sends information to the server about the RFCs it has in the RFC folder.
   2. For demo purpose we have shown 2 clients here:

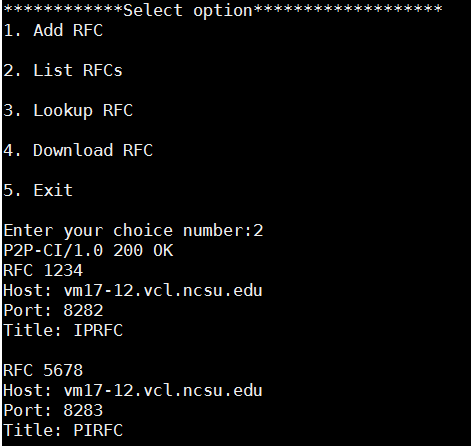




At the same time on the server,

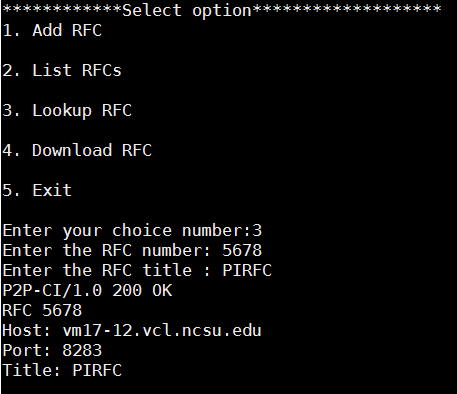


* 1. The client/peer would be prompted with several options to choose from. Option selected = 2 (LIST RFC)

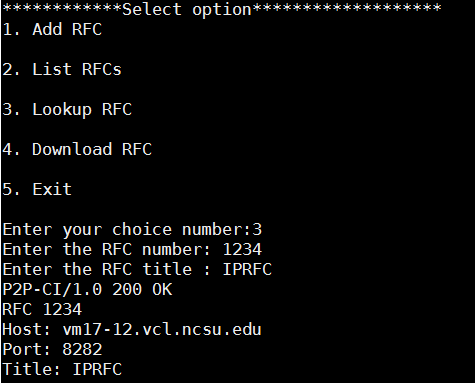


* 1. Option selected = 3 (LOOKUP RFC)

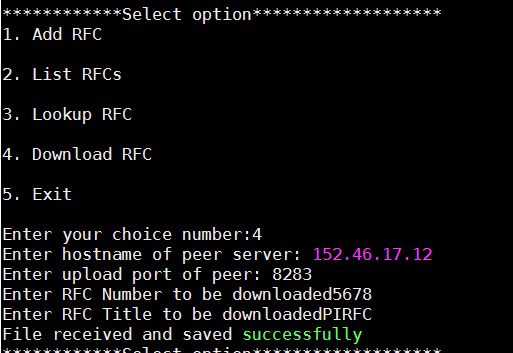
Peer 1 is able to see the Peer 2’s RFC



Similarly, Peer 2 is able to see the Peer 1’s RFC



* 1. Option selected = 4 (DOWNLOAD RFC) Peer 1 is download Peer 2’s RFC – no. – 5678, title – PIRFC



Ultimately you can choose 5 to exit and close the client connection.