NAME : ARPAN MANDAL ROLL : 001910501061 GROUP : A3 CLASS :BCSE III (UG)

ASSIGNMENT 7

Design and implement FTP and DHCP protocol in simulated network environment using socket in python

Problem statement :

Problem Statement :- Implement any two protocols using TCP/UDP Socket as suitable.

1. BOOTP

2. FTP

3. DHCP

4. BGP

5. RIP

FTP

Design: Here, I use python TCP socket to to implement this algorithm,

**Components of the system**

**Client :-** client is responsible for reading the contents from the input file and sends the data to the server following the Transmission Control Protocol, i.e, by first establishing a connection with the server and then transmitting the contents of the file. The client process mainly communicates with the server process via a single port which can be considered as the end point for process communication.

**Server :-** It runs on a specific port with a specific process id and is responsible for receiving the data from the client process. Finally it will write the contents received from the client process in the my\_data.txt file which is saved in server\_data directory.

**FTP :-** FTP stands for File transfer protocol. FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another. It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet .It is also used for downloading the files to computers from other servers.

FILE DATA

SERVER

CLIENT

ACK

SERVER FILE

CLIENT FILE

**CODE:**

**FOR CLIENT-**

import socket

import time

IP='localhost'

PORT=80

ADDR=(IP,PORT)

file\_name="my\_file.txt"

def main():

    client=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

    client.connect(ADDR)

    file=open("client\_data/"+file\_name ,"r") #read particular file we need to send

    data=file.read()

    time.sleep(1)

    client.send(file\_name.encode('utf-8')) #send name of the file to server

    print("[CLIENT] : filename is sent")

    time.sleep(1)

    msg=client.recv(1024).decode('utf-8')

    print(msg)

    time.sleep(1)

    client.send(data.encode('utf-8'))

    print("[CLIENT] : file data sent")

    msg=client.recv(1024).decode('utf-8');

    time.sleep(1)

    print(msg)

    time.sleep(2)

    client.close()

if \_\_name\_\_=="\_\_main\_\_":

    main()

**FOR SERVER-**

import socket

import time

IP=’localhost’

PORT=80

ADDR=(IP,PORT)

def main():

    print(“…..................server started…..................”)

    server=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

    server.bind(ADDR)

    server.listen()

    print(“<<<<istening>>>>”)

    while True:

        conn,addr=server.accept()

        print(“New connection ,”,addr,”connected”)

        filename=conn.recv(1024).decode(“utf-8”) # read the name of the file sent by the client

        time.sleep(1)

        print(“[SERVER] : filename received:”,filename)

        file=open(“server\_data/”+filename,”w”)

        conn.send(“[SERVER] : filename received successfully”.encode(“utf-8”))

        data=conn.recv(1024).decode(‘utf-8’)

        time.sleep(1)

        print(“[SERVER] : data received”)

        file.write(data)

        conn.send(“[SERVER] : data received successfully”.encode(‘utf-8’))

        time.sleep(5)

        conn.close()

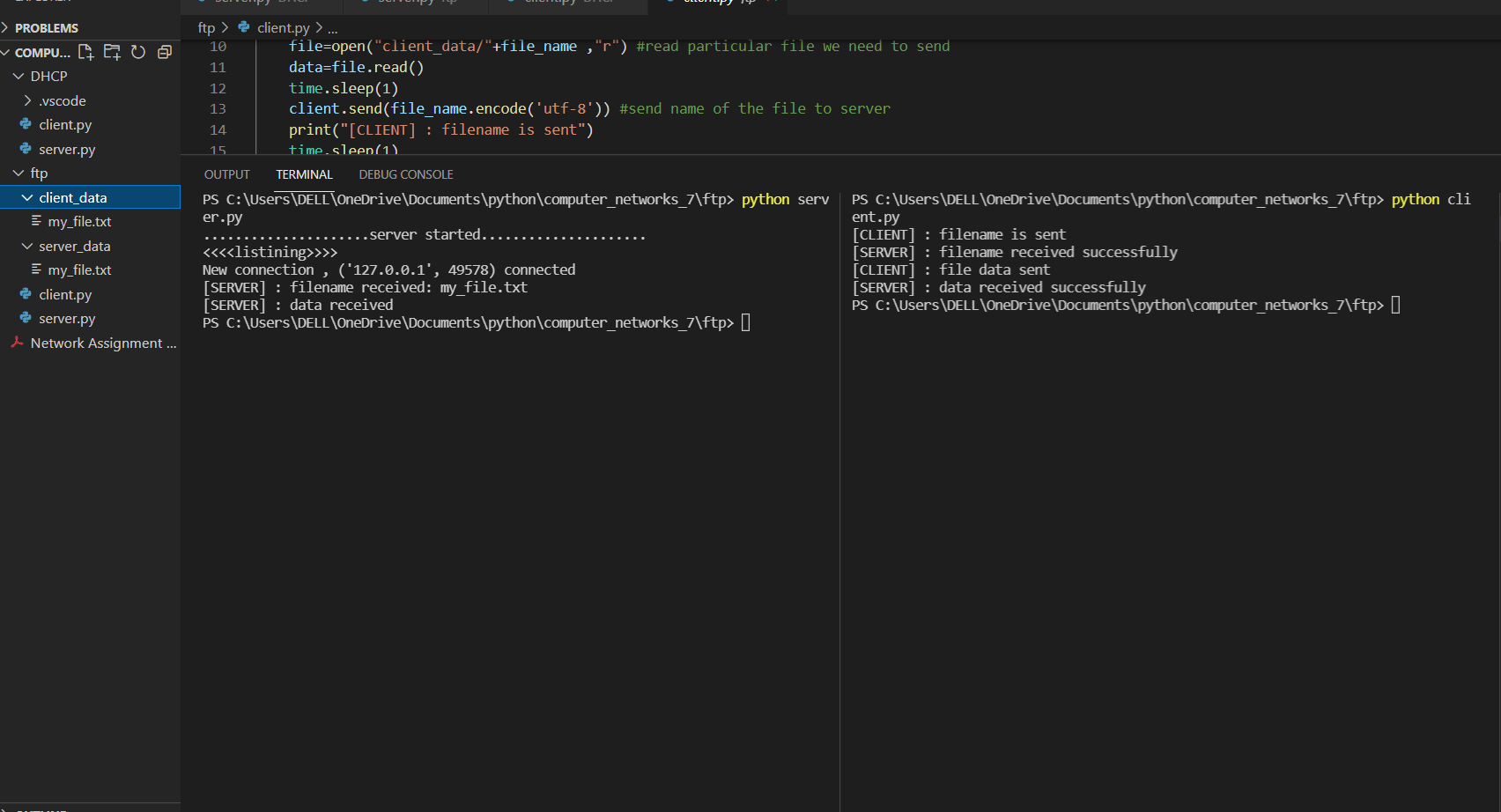
        server.close()

        break

if \_\_name\_\_==”\_\_main\_\_”:

    main()

**OUTPUT-**

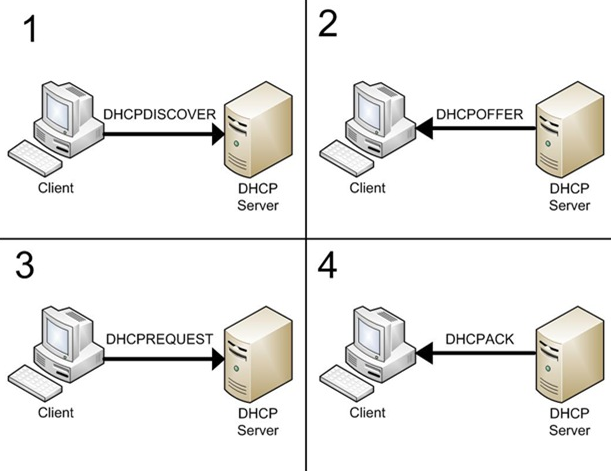


**DHCP**

**System Design Components**

**Client :-** The client process will be created in a particular port with a specific process id and if it’s within range of maximum number of clients that can be accomodated, it will be assigned with an ip address by the DHCP server from the ip address database.

**Server :-** The server process is responsible for allocating an ip address to all valid clients. If a client disconnects itself then it’s ip will be freed and will be available to be assigned for another valid client.



[This Photo](http://www.dailytelecominfo.com/2013/12/sip-interview-questions-answers.html) by Unknown Author is licensed under [CC BY](https://creativecommons.org/licenses/by/3.0/)

**CODE –**

**FOR SERVER-**

import socket

import os

#AF\_INET used for IPv4

#SOCK\_DGRAM used for UDP protocol

s = socket.socket(socket.AF\_INET , socket.SOCK\_DGRAM )

#binding IP and port

print (os.getpid())

available\_ip\_queue = []

available\_ip\_queue.append('0.0.0.0')

available\_ip\_queue.append('0.0.0.1')

available\_ip\_queue.append('0.0.0.2')

connected\_clients = []

s.bind(('127.0.0.1',12345))

print("Server started ...")

print("Waiting for Client response...")

#recieving data from client

#dictionary to handle all the clients

clients = {}

while True:

    data, addr = s.recvfrom(1024)

    dormant\_clients = []

    if addr in clients.keys():

        s.sendto('acknowledgement msg'.encode(), addr)

        print (str (data.decode()) + ' Received from client with ip :- ' + str(clients[addr]))

    else:

        msg = ''

        if len(available\_ip\_queue) == 0:

            msg = 'Sorry client cannot be provided with ip :- '

        else:

            msg = 'hello new client your assigned ip is :- ' + available\_ip\_queue[0]

            clients[addr] = available\_ip\_queue[0]

            available\_ip\_queue.pop(0)

        s.sendto(msg.encode ('utf-8'), addr)

**FOR CLIENT-**

import socket

import os

#client program

s = socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)

process\_id = str(os.getpid())

count = 0

while True:

       #ip ,port = input("Enter server ip address and port number :\n").split()

       print (os.getpid())

       m = input("Enter data to send server: ")

       res = s.sendto(m.encode('utf-8'), ('127.0.0.1',12345))

       data, addr = s.recvfrom(1024)

       if count == 0:

           print (str(data.decode()))

           count = 1

**OUTPUT-**

