Computer Networks Laboratory

Assignment 7

Name: Anirban Das Class: BCSE-III Roll: 001910501077 Group: A3

Problem Statement:

Network, Transport and Application layer protocols.

Implement any two protocols using TCP/UDP Socket as suitable:

1. BOOTP
2. FTP
3. DHCP
4. BGP
5. RIP

Design:

TCP/UDP Sockets:

To manage the connection between application layer network protocols, TCP and UDP use ports and sockets. TCP and UDP operate at the host-to-host layer in the IP communication model and provide host-to-host communication services for the application layer protocol. This means an application layer protocol is on one IP host connecting to an application layer protocol on another IP host.

TCP is a connection-oriented protocol. Connection-orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.

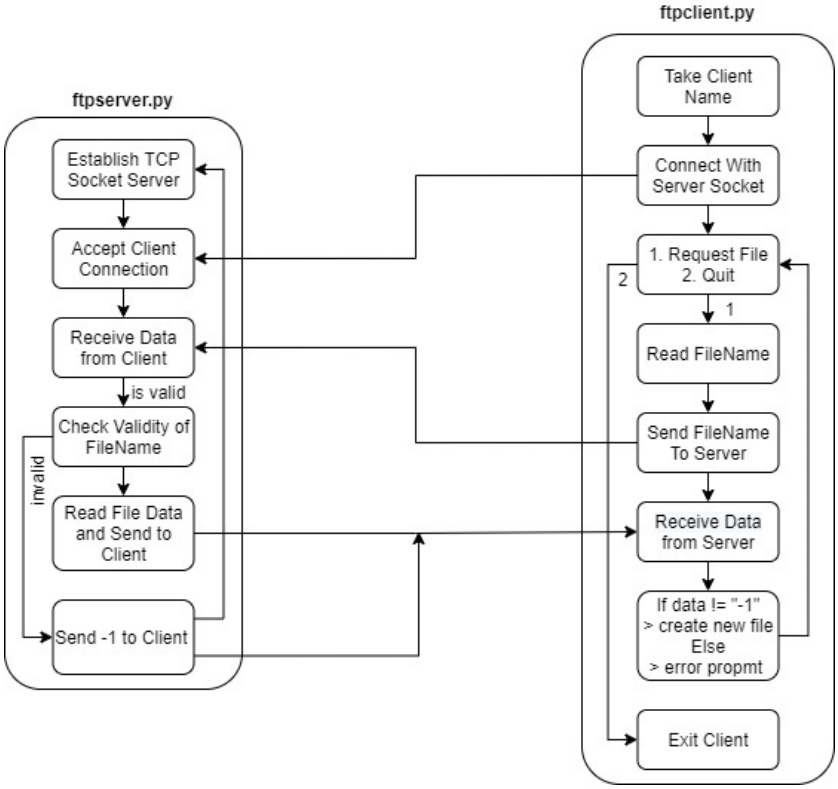
UDP is the Datagram oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, and terminating a connection. UDP is efficient for broadcast and multicast type of network transmission.

PROTOCOLS IMPLEMENTED

File Transfer Protocol:

The **File Transfer Protocol** (**FTP**) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network. FTP is built on a client–server model architecture using separate control and data connections between the client and the server.

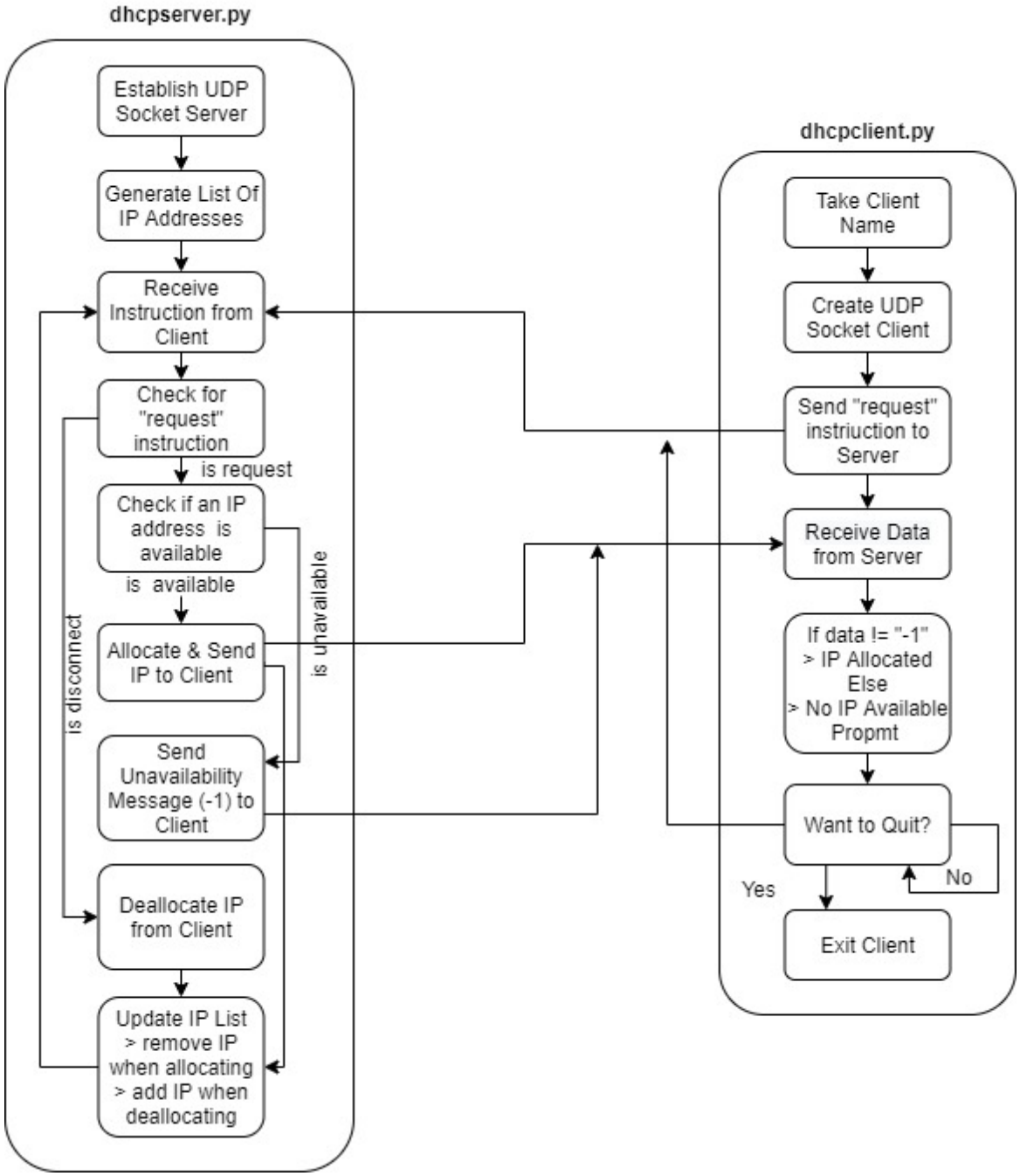
TCP Sockets are used to implement the client-server model in File Transfer Protocol. The following flow diagram explains the control flow of the algorithm:



Dynamic Host Control Protocol:

The **Dynamic Host Configuration Protocol** (**DHCP**) is a network management protocol used on Internet Protocol (IP) networks for automatically assigning IP addresses and other communication parameters to devices connected to the network using a client–server architecture.

UDP Sockets are used to implement the client-server model in File Transfer Protocol. The following flow diagram explains the control flow of the algorithm:



Implementation:

Dynamic Host Control Protocol

server

dhcp-udp > dhcpserver.py > ...

```python
import socket
import sys

# constants
HOST = '127.0.0.1'      # Standard loopback interface address (localhost)
OWNport = 65100         # Port to listen on (non-privileged ports are > 1023)
commonGateway = '127.0.0.'
maxIPs = 2              # at max 100 clients can connect


# Creating a UDP Socket
server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
server.bind((HOST, OWNport))
print("DHCP Server Started!")

# 100 temporary ip's starting from 127.0.0.3 to 127.0.0.102
ipList = [i for i in range(3, 3 + maxIPs)]

# dictionary to map current client's address with
# temporary IP address generated by DHCP
allocatedClientIP = dict()
cnt = 0

while True:
    message, address = server.recvfrom(1024)
    message = message.decode()

    if message == 'request':
        print("Temporary IP Requested...")

        if(len(ipList) > 0):
            temp = ipList.pop()  # getting an available IP
            tempAddress = commonGateway + str(temp) # generating temporary IP
            allocatedClientIP[tempAddress] = temp  # updating the current client dict
            cnt += 1
            print("Temporary IP [" + tempAddress + "] allocated!")
            print("Currently " + str(cnt) + " clients are online\n")
            server.sendto(str.encode(tempAddress), address)
        else:
            print("No IP Address available from Server, maximum clients connected..\n")
            server.sendto(str.encode("-1"), address)

    else:
        print("IP Deallocation Requested...")
        temp = allocatedClientIP[message]
        print("Deallocating IP [" + message + "]...")
        allocatedClientIP.pop(message) # deleting this from current client dict
        ipList.append(temp) # making this IP available again
        cnt -= 1
        print("Deallocated IP [" + message + "]!")
        print("Currently " + str(cnt) + " clients are online\n")

```

client

dhcp-udp > dhcpclient.py > ...

```python

name = input("Enter name of the client: ")

# Create a UDP Socket
client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
print("Client Started...")

address = (HOST, DHCPport)

print("Requesting for temporary IP from DHCP Server...")
client.sendto(str.encode("request"), address)
ip, server = client.recvfrom(1024)
if(ip.decode() == "-1"):
    print("\nNo IP Address available from Server, maximum clients connected..\n")
else:
    print("Temporary IP Allocated! [" + ip.decode() + "]")
    while True:
        print("Press q to quit, anything else to stay online.")
        choice = input()
        if choice == "q" or choice == "Q":
            break
    print("Terminating...\n")
    client.sendto(ip, address)

```

File Transfer Protocol

server

ftp-tcp > server > ftpserver.py > ...

```python
import socket
from pathlib import Path

HOST = '127.0.0.2'  # Standard loopback interface address (localhost)
PORT = 65300        # Port to listen on (non-privileged ports are > 1023)

print("FTP Server started!!")
while True:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1) # to avoid 'Address already in use error'

    s.bind((HOST, PORT))
    print("Listening for a connection on its own port....\n")
    s.listen()

    conn, addr = s.accept()
    name = conn.recv(1024).decode()
    filename = conn.recv(1024).decode()
    print("Client ", name, " with address " , addr , " is requesting file: ", filename)

    # checking file existence
    filename = Path(filename)
    if filename.is_file():
        file = open(filename, 'r')
        data = file.read()
        conn.send(bytes(data, "utf-8"))
        print("Data sent!\n")

    else:
        error = "File does not exists. Terminating Request..."
        print(error, "\n")
        conn.send(bytes("-1", "utf-8"))

    s.close()
    print("FTP Server still running!")

```

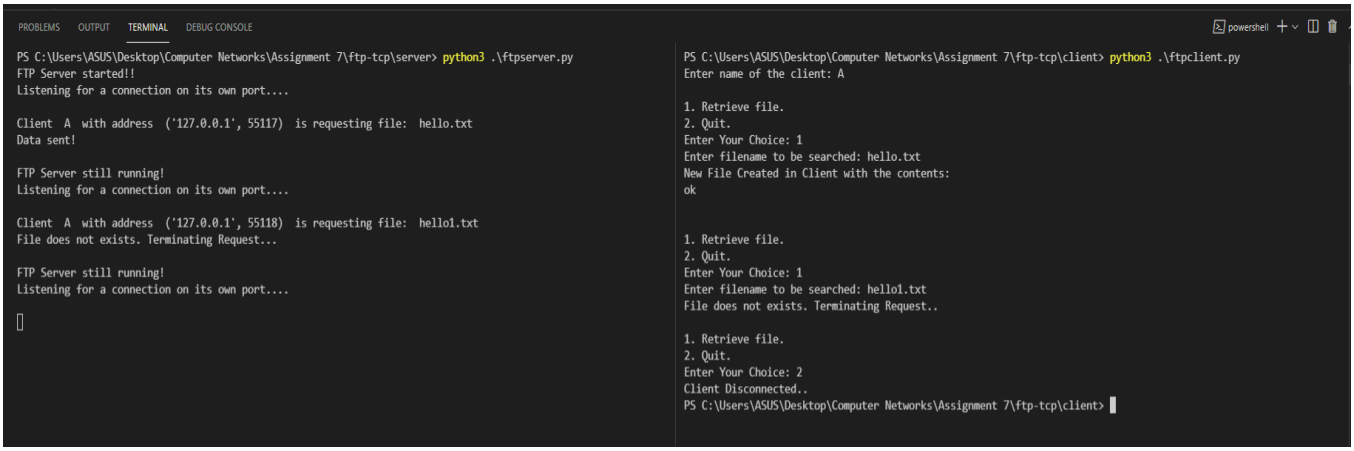
client

p-tcp > client > ftpclient.py > ...

```python
import socket

Host = '127.0.0.2'  # The server's hostname or IP address
FTPport = 65300        # The port used by the server

name = input("Enter name of the client: ")

while True:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    choice = int(input("\n1. Retrieve file.\n2. Quit.\nEnter Your Choice: "))
    if choice == 1:
        s.connect((Host, FTPport))
        s.send(bytes(name, "utf-8"))
        filename = input("Enter filename to be searched: ")
        s.send(bytes(filename, "utf-8"))
        data = s.recv(1024).decode()

        # if filename is invalid
        if data == "-1":
            print("File does not exists. Terminating Request..")
            continue
        else:
            print("New File Created in Client with the contents: \n" + data + "\n")
            file = open(filename, "w")
            file.write(data)

        s.close()

    elif choice == 2:
        print("Client Disconnected..")
        break
    else:
        print("Invalid Choice. Try Again.")
```

Results & Output:

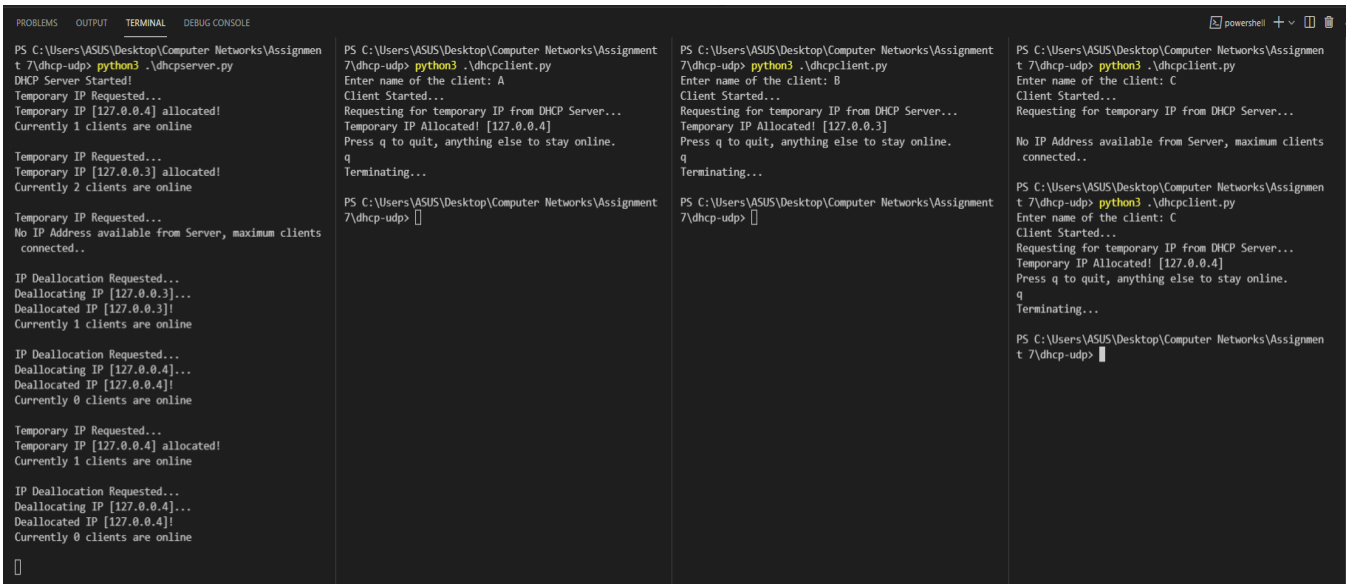
File Transfer Protocol



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                                   powershell

PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\ftp-tcp\server> python3 .\ftpserver.py
FTP Server started!!
Listening for a connection on its own port....

Client  A  with address  ('127.0.0.1', 55117)  is requesting file:  hello.txt
Data sent!

FTP Server still running!
Listening for a connection on its own port....

Client  A  with address  ('127.0.0.1', 55118)  is requesting file:  hello1.txt
File does not exists. Terminating Request...

FTP Server still running!
Listening for a connection on its own port....
```

```
PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\ftp-tcp\client> python3 .\ftpclient.py
Enter name of the client: A

1. Retrieve file.
2. Quit.
Enter Your Choice: 1
Enter filename to be searched: hello.txt
New File Created in Client with the contents:
ok


1. Retrieve file.
2. Quit.
Enter Your Choice: 1
Enter filename to be searched: hello1.txt
File does not exists. Terminating Request..

1. Retrieve file.
2. Quit.
Enter Your Choice: 2
Client Disconnected..
PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\ftp-tcp\client>
```

Dynamic Host Control Protocol



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                                   powershell

PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp> python3 .\dhcpserver.py
DHCP Server Started!
Temporary IP Requested...
Temporary IP [127.0.0.4] allocated!
Currently 1 clients are online

Temporary IP Requested...
Temporary IP [127.0.0.3] allocated!
Currently 2 clients are online

Temporary IP Requested...
No IP Address available from Server, maximum clients
 connected..

IP Deallocation Requested...
Deallocating IP [127.0.0.3]...
Deallocated IP [127.0.0.3]!
Currently 1 clients are online

IP Deallocation Requested...
Deallocating IP [127.0.0.4]...
Deallocated IP [127.0.0.4]!
Currently 0 clients are online

Temporary IP Requested...
Temporary IP [127.0.0.4] allocated!
Currently 1 clients are online

IP Deallocation Requested...
Deallocating IP [127.0.0.4]...
Deallocated IP [127.0.0.4]!
Currently 0 clients are online
```

```
PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp> python3 .\dhcpclient.py
Enter name of the client: A
Client Started...
Requesting for temporary IP from DHCP Server...
Temporary IP Allocated! [127.0.0.4]
Press q to quit, anything else to stay online.
q
Terminating...

PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp>
```

```
PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp> python3 .\dhcpclient.py
Enter name of the client: B
Client Started...
Requesting for temporary IP from DHCP Server...
Temporary IP Allocated! [127.0.0.3]
Press q to quit, anything else to stay online.
q
Terminating...

PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp>
```

```
PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp> python3 .\dhcpclient.py
Enter name of the client: C
Client Started...
Requesting for temporary IP from DHCP Server...

No IP Address available from Server, maximum clients
 connected..

PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp> python3 .\dhcpclient.py
Enter name of the client: C
Client Started...
Requesting for temporary IP from DHCP Server...
Temporary IP Allocated! [127.0.0.4]
Press q to quit, anything else to stay online.
q
Terminating...

PS C:\Users\ASUS\Desktop\Computer Networks\Assignment 7\dhcp-udp>
```

Comments:

This assignment helped me understand the working procedure of FTP and DHCP protocols and successfully implementing them using TCP/UDP sockets in Python 3.