GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

Experiment No: 2

Socket Programming UDP using Python

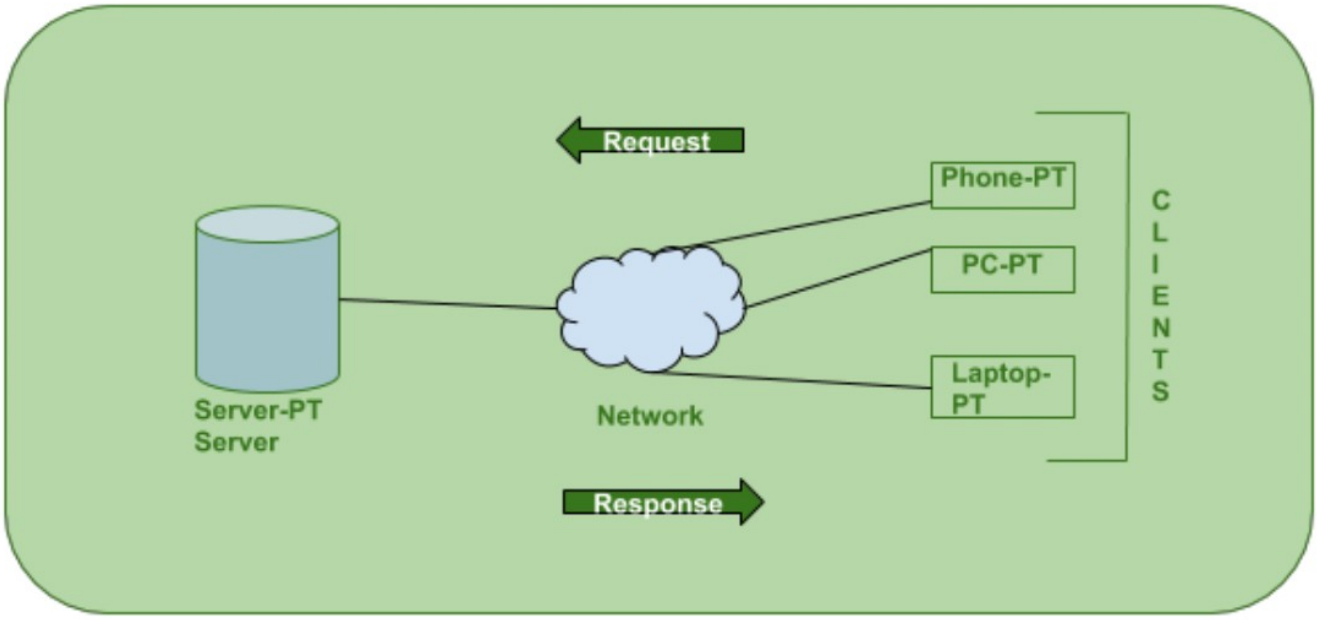
Aim: To implement socket programming using UDP connection between client and server.

Theory:

The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.

Client: When we talk the word Client, it means to talk of a person or an organization using a particular service. Similarly in the digital world a client is a computer (Host) i.e., capable of receiving information or using a particular service from the service providers (Servers).

Servers: Similarly, when we talk the word Servers, it means a person or medium that serves something. Similarly in this digital world a Server is a remote computer which provides information (data) or access to particular services.



GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

Program:

1) Client side import socket

```
msgFromClient = ""
bytesToSend = ""
serverAddressPort = ("127.0.0.1", 8090) bufferSize = 1024

def UserInput():
msgFromClient=input("\nEnter message for server") bytesToSend=str.encode(msgFromClient)
UDPClientSocket.sendto(bytesToSend, serverAddressPort)

# Create a UDP socket at client side
UDPClientSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)

# Send to server using created UDP socket

while(True):
UserInput()
msgFromServer = UDPClientSocket.recvfrom(bufferSize) msg = "Message from Server
{}".format(msgFromServer[0]) print(msg)
```

2) Server side

```
import socket
localIP = "127.0.0.1" localPort = 8090 bufferSize = 1024

msgFromServer = "Hello UDP Client" bytesToSend = str.encode(msgFromServer)

# Create a datagram socket
socketUDP = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

# Bind to address and ip socketUDP.bind((localIP, localPort))

print("UDP server up and listening")

# Listen for incoming datagrams while(True):

bytesAddressPair = socketUDP.recvfrom(bufferSize) message = bytesAddressPair[0]
address = bytesAddressPair[1]
clientMsg = "Message from Client:{}".format(message) clientIP = "Client IP Address:{}".format(address)

print(clientMsg,"\n")

print(clientIP,"\n")
# Sending a reply to client socketUDP.sendto(bytesToSend, address)
```

Output:

```
aditya@Aditya001:~$ cd nsexpts/
aditya@Aditya001:~/nsexpts$ python3 server.py
UDP server up and listening
Message from Client:b'hello'
Client IP Address:('127.0.0.1', 58547)
Message from Client:b'hello aditya'
Client IP Address:('127.0.0.1', 58547)
Message from Client:b'hello'
Client IP Address:('127.0.0.1', 58547)
Message from Client:b'hello aditya'
Client IP Address:('127.0.0.1', 58547)
```

```
aditya@Aditya001: ~/nsexpts
aditya@Aditya001:~$ cd nsexpts/
aditya@Aditya001:~/nsexpts$ python3 client.py
enter the message:
 hello
Message from Server b'Hello UDP Client'
enter the message:
 hello aditya
Message from Server b'Hello UDP Client'
enter the message:
 hello
Message from Server b'Hello UDP Client'
enter the message:
 hello aditya
Message from Server b'Hello UDP Client'
enter the message:
```

```
aditya@Aditya001: ~
aditya@Aditya001:~$ sudo tcpdump -A -i lo
[sudo] password for aditya:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
11:17:58.268655 IP localhost.58547 > localhost.8090: UDP, length 5
E..!.p@.@.XY.............. hello
11:17:58.269036 IP localhost.8090 > localhost.58547: UDP, length 16
E..,.q@.@.XM.............+Hello UDP Client
11:18:41.576025 IP localhost.58547 > localhost.8090: UDP, length 12
E..(..@.@.R.............'hello aditya
11:18:41.576302 IP localhost.8090 > localhost.58547: UDP, length 16
E..,..@.@.R.............+Hello UDP Client
^C^C
4 packets captured
8 packets received by filter
0 packets dropped by kernel
aditya@Aditya001:~$
```

Conclusion: Socket Programming UDP using python was successfully implemented.