EXP -

DATE - 15/10/2024 ECHO CLIENT

AIM - To implement echo client server using TCP/UDP socket

Algorithm

server.py :

→ create a UDP socket
→ Bind the socket to specific IP address (127.0.0.1) & port (12345)
→ continious listen for incoming message
→ When message received decode it
→ Display message along with sender address
→ Repeat Infinitely

client.py

→ create UDP socket
→ Set a timeout for socket to avoid waiting
→ Send a predefined message hellow to server IP address & port 12345
→ If no response received in timeout period, print timeout message
→ close socket after sending message

CODE:

server.py

```
import socket
def start_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET
                       socket.SOCK_DGRAM)
        s.bind((host,port))
```

```
while True:
    data, addr = s.recvfrom(1024)
    print("recieved message from", addr, ":", data.decode())

if __name__ == "__main__":
    start_server()
```

client.py

```
def ping(serverHost='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.settimeout(5)

        try:
            s.sendto(b'Hello', (host, port))
            print("message sent to server")
        except socket.timeout:
            print("Request timed out")
if __name__ == "__main__":
    ping_server()
```

OUTPUT

server.py

Terminal

> python server.py

>> UDP server running on 127.0.0.1:12345

client.py

Terminal

> python client.py

>> message sent to server

server terminal:

Received message from ('127.0.0.1', 56003) : Hello

Result - Thus the program echo client server using UDP socket has been implemented & executed successfully

EXNO - 12B
DATE - 11/11/2024

CLIENT SERVER

AIM

Implement chat client server using TCP/UDP sockets

CODE

server.py

```
import socket
    def recvl()
      port=12345
      host='127.0.0.1'
      with socket.socket(socket.
          AF_INET (socket
          SOCK_DGRAM) as S:

  s.send((Chost,port))
while(True):
    d,addr = s.recv.form(1024)
    print("client", (d.decode))
    a=input("write here reply")
    s.send.to(a.encode(), addr)
    if(a=='end')
        break
        exit
    recvl()
```

client.py

```
import socket
import time
def recv2(a):
    host = '127.0.0.1'
    port = 12345
with socket.socket(socket.AF_INET,
        socket.SOCK_DGRAM) as s
s.sendto(a.encode(), (host, port))
data, addr = s.recvfrom(1024)
print('{}'.format(data.decode()))
while(True):
    a = input("Enter message")
    if(a == "end"):
        recv1(a)
        break

    else:
        recv2(a)
```

Result:

Thus client server was implemented successfully using TCP/UDP socket