Ex - 12a)

Date : 10/10/24 TCP/UDP Sockets

AIM:

To implement echo client server using TCP/UDP Sockets.

ALGORITHM:

Server.py :

→ Create a UDP Socket

→ Bind the socket to specific IP address (127.0.0.1) & port (12345)

→ Continuous 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 hello 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) as
        s.bind((host, port))
        print(f"UDP server running on
                {host}:{port}")
        while True:
            data, addr = s.recvfrom(1024)
            print(f"received message from
                {addr}: decoded {}")
if __name__=="__main__":
    start_server()
```

Client.py:

```
def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET,
            socket.SOCK_DGRAM) as s:
        s.timeout(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 of ~~eac~~ echo client server using UDP sockets has been implemented & executed successfully.