Ex: 12a Implement echo client
Date: 10/10/24 server using 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.
→ continuously 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:

import
def
def
with
except

echo client

TCP/UDP

client

IP

incoming

decode

to avoid

hallow

1234

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:
        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}: {data.decode()}")
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 time out")
if __name__ == "__main__"
    ping_server().
```

output:

server.py

terminal

>python server.py

>>

UDP server running on 127.0.0.1: 123

client.py

terminal

> python client.py

>>

message sent to server.

server terminal:

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

Result:

Thus the program of echo client server using UDP sockets has been implemented & executed sucessfully

EXP: 13

Date: 11/10/24

Aim:

To impl

Algorithm:

UDP server:

→ crea

specific ad

→ wait

→ proc

→ send

UDP clien

→ crea

timeout.

→ se

→ If

response &

→ If

print rea