Exper: 12.0
Date: 18/10/24

Echo client & server using TCP/UDP sockets

Aim:

To implement echo client server using TCP/UDP sockets.

Algorithm

Server.py

→ Create a UDP socket
→ Bind 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 timeout for socket to avoid waiting
→ Send predefined message bellow to server IP address & port 12345
→ 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_INE
        socket.SOCK_DGRAM) as s
```

```
    s.bind((host, port))
    while True:
        data, addr = s.recvfrom(1024)
        print(f"Received message from
{addr}: {data.decode()}")
if __name__ == "__main__":
    startserver()
```

client.py

```
def ping_server(host = '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()
```

o/p:

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 echo cline server using UDP sockets has been implemented & executed successfully.