

Aim

to implement echo client server using TCP Sockets

Algorithm

Server.py:

⇒ Create a UDP socket

⇒ Bind the socket to specific address

(127.0.0.1) & port (12345)

⇒ continuously listen for incoming message

⇒ when message received - decode

⇒ 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 Port 12345

Code

server.py

```
import socket
```

```
def start_server(host="127.0.0.1", port=12345):  
    with socket.socket(socket.AF_INET,  
                        socket.SOCK_STREAM)  
    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}")
```

```
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_STREAM)  
    s: timeout(s)
```

~~Try~~

```
s.sendto(b'Hello', (host, port))  
print("message sent to server")  
except socket.timeout:
```

```
    print(" ")
```

Output

Server.py

> python Server.py

>> UDP Server running on 127.0.0.1:12345

Client.py

> python Client.py

>> message sent to Server

Server terminal:

Receiver message from ('127.0.0.1', 8600)

Result

thus the programming of echo client server using
UDP sockets has been implemented & executed