Ex
24/9/24

Echo Client Server using TCP/UDP

Aim:

To implement echo client server using TCP/UDP socket

Algorithm:

Server.py:

Create a UDP socket

Print the socket to specific IP address & port (12345)

Continuously listen for incoming message

When message received - decode it

Repeat indefinitely

Client.py

Create UDP socket

Set a timeout for socket to avoid waiting

Send a predefined message to server IP address & port 12345

If no response received in time / Print time out 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:
    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.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): b'Hello'

Result:

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