

Exp No: 12a
22/9/23

Practical - 12a
End-to-End communication at Transport layer

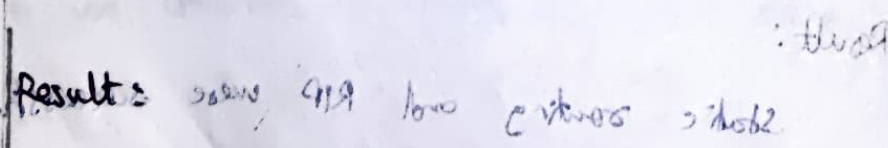
Aim: a) Implement echo client server using TCP/UDP sockets

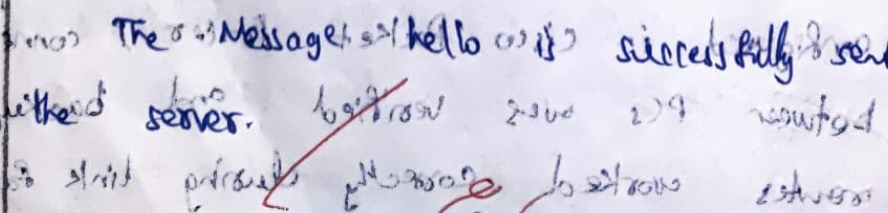
Algorithm:

```
import socket
import time

def ping_server(host = '127.0.0.1', port = 10345):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        try:
            s.sendto(b'Hello', (host, port))

        except s.timeout:
            print("Request timed out")
    ping_server()
```

Result:  The screenshot shows a terminal window with the command 'ping 127.0.0.1' and the output 'Pinging 127.0.0.1: 32ms: success!'. This indicates that the network connectivity is working.

The message 'Hello' is successfully sent to the server.  The screenshot shows a terminal window with the command 'nc 127.0.0.1 10345' and the output 'Hello'. This indicates that the echo server is working correctly.

14/10/23

22/9/23

Implement chat client over using TCP/UDP

b) Aim: Implement chat client using TCP/UDP sockets

Algorithm:

import socket

def start_server(host = '127.0.0.1', port = 12345)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server as s:

s.bind((host, port))

s.listen(1)

while True:

data, addr = s.recv(1024)

print(f"Received Message from {addr}: {data.decode('utf-8')}")

s.close()

output:

UDP server running on 127.0.0.1:12345

Received message from (127.0.0.1, 52345): Hello

(127.0.0.1, 52345)

("Hello")

Result:

This implementation of chat client server

using TCP/UDP socket performed successfully.

Handwritten signature