CS23532-COMPUTER NETWORKS-LAB MANUAL
Practical -12

Name: Tharunraj
RegNo:230701362

AIM: - a) Implement echo client server using TCP/UDP sockets.

Algorithm for Server :

1. Create a TCP/UDP socket, bind it to a specific IP address and port, and start listening for client connections.
2. Accept the client connection and receive the message sent by the client.
3. Send the same message back to the client (echo) and close the connection.

Algorithm for Client :

1. Create a TCP/UDP socket and connect it to the server using its IP address and port number.
2. Send a message to the server and wait for the response.
3. Receive the echoed message from the server and close the connection.

Program:

client.py

```
import socket
import time

def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        try:
            s.sendto(b'Hello', (host, port))
        except s.timeout:
            print("Request timed out")

ping_server()
```

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()}")


start_server()
```

Output(server.py):

UDP Server running on 127.0.0.1:12345
Received from ('127.0.0.1', 55342): Hello Server
Received from ('127.0.0.1', 55342): How are you?
Received from ('127.0.0.1', 55342): Bye

Output(client.py):

Enter message to send: Hello Server
Received from server: Hello Server

Enter message to send: How are you?
Received from server: How are you?

Enter message to send: Bye
Received from server: Bye

RESULT:

Program for implementing echo client server using TCP/UDP sockets has been executed successfully.