Exp No: 12 End- End communication at transport Layer

AIM:
To implement echo client server using TCP / UDP Sockets.

Program:

```
import socket
import time
import sys
# UDP. Ping (echo) server
def - udp - ping - server (host = "127.0.0.1", port=200)
server_socket = socket.socket (socket.AF_INET,
                               socket.sock_DGRAM)
server_socket.bind ((host, port))
while True:
    msg. client_address = server_socket.receive
                                    from (1024)
    print (f"[server] received {msg.decode()}
    from {client_address}")
# UDP. ping (Echo) client
    client_socket.settimeout (1)
    for i in range (1, count+1):
        start = time.time()
        client_socket.sendto (msg.encode()
            server_host, server_port))
try:
    data = client_Socket.recvfrom (1024)
    end = time.time()
    rtt = (end - start) * 1000.
```

except s

if name =
if

els

Input:
python

Output:
Server
[Server

Client
Reply
RTT =

Reply
RTT =

Result:

Server

```
        except socket.timeout :
            print(f"Request {i+1} timeout")

if __name__ == "__main__":
    if len(sys.argv) > 1 and sys.argv[1] == "server":
        udp_ping_server()
    else:
        udp_ping_client()
```

Input :

```
python udp_ping_program.py
```

Output :

Server side output :

```
[Server] Listening on 127.0.0.1 : 12000
```

Client side output :

```
Reply from 127.0.0.1200 | Ping | 1738842178|
RTT = 1.23 ms
Reply from 127.00.1200 | Ping B 1738842178|
RTT = 1.10 ms.
```

Result :

Therefore implementation of echo client server using TCP / UDP sockets is executed.

22/1 10/10