06/11/24 Experiment -12

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

TCP SERVER

```
import socket

def tcp-echo-server():
    server-socket = socket.socket (socket.AF_INET,
                        socket.SOCK_STREAM)
    server-socket.bind ("127.0.0.1", 65432))
    server-socket.listen()

    print ("TCP server listening")
    While True:
        conn, addr = server-socket.accept()
        print(f" Connected by {addr}")
        While True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
        conn.close()
if __name__ == "__main__":
    tcp-echo-server()
```

TCPCLIENT

```
import socket

def tcp-echo-client ():
    client_socket= socket.socket(socket.AF_INET,
                            socket.SOCK_STREAM)
    client_socket.connect(("127.0.0.1", 65432))

    message =input("Enter message:")
    client_socket.sendall message.encode())

    data= client_socket.recv(1024)
    print("Recieved :", data.decode())
    client_socket.close()
if __name__ =="__main__":
    tcp-echo_client()
```

UDP SERVER

```
import socket
def udp-echo-server():
    server-socket = socket.socket(socket.AF_INET
                        socket.SOCK_DGRAM)
    server_socket.bind(("127.0.0.1", 65432))
    print("UDP server listening...")
    while True:
        data,addr= server_socket.recvfrom
        print(f"Recieved from{addr}:{data.deco
        server-socket.sendto(data,addr)

if __name__ =="__main__":
    udp-echo-server()
```

UDP CLIENT

```
import socket

def udp_echo_client():
    client_socket = socket.socket(socket.AF_INET,
                                  socket.SOCK_DGRAM)
    message = input(" Enter message to send")
    client_socket.sendto(message.encode(),
                         ("127.0.0.1", 65432))
    data, _ = client_socket.recvfrom(1024)
    print("Recieved:", data.decode())
    client_socket.close()

if __name__=="__main__":
    udp_echo_client()
```

INPUT 1

Hello!

OUTPUT 2

TCP server listening....

Recieved : Hello!

INPUT 2

Hello:

OUTPUT 2

UDP Server Listening:

Recieved : Hello!

Aim b) chat client using TCP/UDP

TCP SERVER

```
import socket
import threading

def handle-client (client-socket, client- address):
    print f"New connection from {client address}")
    While True:
            message = client_socket.recv(1024).deco
            if messag:
                print(f"{client address}: {message}")
                broadcast_message (f"{client_address}:
                                {message}", client_sock)
          else:
             break
    except:
         break
    client_socket.close()
def broadcast_message (message, sendersocket):
    for client in clients:
          if client != sender_socket:
               client.send(message.encode())

def: tcp_chat_server ():
    Server_socket = socket.socket(socket.AF_INET,
                        socket.SOCK_STREAM)
    Server_socket.bind(("127.0.0.1", 65432))
    Server_socket.listen()
    print ("TCP chat server listening...")
```

```
While True:
        client_socket client_address server_socket
        client.append(client_socket)
        threading.Thread
```

TCP CLIENT

```
import socket
import threading

def recieve_messages(client_socket):
        while True:
            try:
                message= client_socket.recv(1024).decod
                if message:
                        print(\n+ message)

            except :
                    print("Disconnected from server")
                    break

def tcp_chat_client()
        client_socket= socket.socket(socket.AF_INET
                                            socket_sock )
        client_socket.connect(("127.0.0.1", 65432))

        while True:
            message = input("You: ")
            client_socket.send(message.encode())

if __name__ == "main__":
            tcp_chat_client().
```

[illegible]