

## TCP/UDP SOCKETS

AIM:

a) Implement echo client server using TCP/UDP Sockets.

client:

```
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 socket.timeout:
```

```
        print("Request timed out")
```

```
if __name__ == "__main__":
```

```
    ping_server()
```

Server:

```
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}")
```

```
    while True:
```

```
        data, addr = s.recvfrom(1024)
```

```
        print(f"Received message from {addr}:
              {data.decode()}")
```

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

OUTPUT: python server.py

UDP server running on 127.0.0.1:12345

Received message from ('127.0.0.1', 59290): hello

python client.py

Received reply from server: Hello, client.

b) Implement chat client server using tcp/UDP sockets:

chat\_serv.py

```
import socket
```

```
def sent():
```

```
    port = 12345
```

```
    host = '127.0.0.1'
```

```
    with socket.socket(socket.AF_INET, socket
```

```
        SOCK_DGRAM) as s:
```

```
        s.bind((host, port))
```

```
        while True:
```

```
            d, addr = s.recvfrom(1024)
```

```
            print("client", {d.decode()})
```

```
            a = input("Enter reply")
```

```
            s.sendto(a.encode(), addr)
```

```
            if a == "end":
```

```
                break
```

```
            exit
```



receive.py

```
import socket
import time
def receive(a):
    host = '127.0.0.1'
    port = 12345
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.sendto(a.encode(), (host, port))
    d, addr = s.recvfrom(1024)
    print(d.decode())
    while True:
        a = input('enter message')
        if (a == "end"):
            receive(a)
            break
        else:
            receive(a)
```

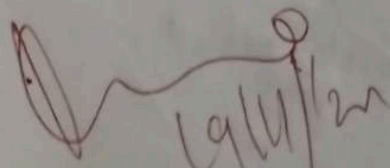
OUTPUT:

python.\chat\_serv.py  
client {'hi'}  
Enter reply: hello  
client {'how are you'}  
Enter reply: I'm fine

python.\recv.py  
Enter message hi  
{'hello'}  
Enter messages how are  
you {'I'm fine'}  
Enter message

RESULT:

Thus, the program is executed successfully & the output is verified.

  
19/11/21