

Ex.NO:12.a)

ECHO CLIENT SERVER

Date

Aim:

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("Hello", (host, port))
```

```
    except s.timeout:
```

```
        print("Request time 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("UDP server running on [host, port]")
```

```
while True:  
    data, address = socket.recvfrom(1024)  
    print(f"Received message from {address[0]}  
          {data.decode()}" )  
  
    if name == "match":  
        grant_server()
```

Output:

Python memory

UDP server running on 127.0.0.1:12345

Received message from ('127.0.0.1', 5927) 14010

1400

Python client.py

Received reply from server: Hello, client

~~8th~~ observations. 1) of Dec 22.

Result:

This is the implementation of echo client server using TCP/UDP sockets is successfully executed and verified.