EX NO: 13
DATE: 22/10/24

Practical-13.

AIM:
Implement your own Ping Problem.

ALGORITHM:-
Server side (UDP)
1. create and bind a UDP socket to a host and port
2. Listen for incoming messages in a loop
3. When a message is received echo it back to the sender.
4. Repeat.

Client side (UDP)
1. create a UDP socket and set a timeout.
2. For each ping.
   - Record the start time
   - Send a ping message to server
   - If no response print "Request Time out"
3. Repeat for a specified number of ping

SOURCECODE:-
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()}")

```
        S.sendto (b'Pong', addr)
if __name__ == "__main__":
start_server()
```

client.py:

```
import socket
import time
def ping_server(host = '127.0.0.1', port=12345

ass:try:
        s.settimeout(2)
        start = time.time()
        s.sendto (b'Ping', (host,port))
        data, addr = s.recvefrom(1024)
        end = time.time()
        Print(f "Received {data.decode} from {addr}
          in {end - start :.2f} seconds")
    except socket.timeout
        Print ("request time out")
if __name__ == "__main__":
    Ping Server()
```

OUTPUT

```
UDP Server Running on 127.0.0.1: 12345
Received message from ('127.0.0.1', 54716): Ping
Received message from ('127.0.0.1', 48076): Ping
Received message from ('127.0.0.1', 48702): Ping.
```

RESULT:

Thus ping program is implemented and Executed successfully.