Ex-13 [redacted]
Date: 19/10/24

[redacted]

AIM:
To implement your own ping program [redacted].

Algorithm:

UDP Server :
→ Create UDP Socket & bind it to a specific address & part
→ Wait for message
→ print message & client address.
→ Send back ping to client.

UDP client :
→ Create UDP socket & set a 2 sec timeout
→ Send 'ping' to server
→ If a respone (ping received - print response & calculate RTT)
→ If no response within 2 sec print request Time out.

Code :

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.received (1024)
            print (f"Received message from
 {addr}: {data.decode ()}")
          s.sendto (b'pong', addr)
if __name__ == "__main__" :
        start_server ()
```

client.py :

```
import time
import socket
def ping_server (host='127.0.0.1',
            port = 12345)
    with socket.socket (socket.AF_INEt,
          Socket.SOCK_DGRAM) as S:
     try :
          s.settime (2)
          start = time.time ()
          s.sendto (b'ping', ((host, port))
          data. addr = s.recv from (1024)
          end = time.time()
```

```
        print(f"Received {data.decode()} from {addr} in {end - start}.2f} seconds");
    except socket.timeout():
        print("Request timed out")
if __name__ == "__main__":
        ping-server()
```

OUTPUT:

```
   Terminal                                Terminal
> python server.py
  UDP server running
   on 127.0.0.1:12345                > python client.py
                                     Received ping
                                      from (127.0.0.1,
Received message from                (12345) in 0.00
('127.0.0.1', 50001: ping               seconds),
```

RESULT:

Thus a ping program has been executed successfully.

[illegible]