25|10|24

EXERCISE 13

Aim:

To implement your own ping program

Algorithm:

UDP server

→ Create UDP socket and bind it to a specific address & port.

→ wait for message.

→ print message & client's address.

→ send back pong to client

UDP client

→ Create UDP socket and set a 2 sec timeout

→ send ping to server.

→ If a response ("Pong is received - print response and 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_NET, 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()
```

<u>client.py</u>

```
import time
import socket
def ping_server(host='127.0.0.1', port=12347):
    with socket.socket(socket.AF_INET, socket.SOCK_
                                    DGRAM) as s:
        try:
            s.settime(2)
            start = time.time()
            s.sendto(b'png', (host, port))
            data, addr = s.recvfrom(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

```
>Python server.py
UDP server running on
  127.0.0.1:12345


Received message from
('127.0.0.1', 50061: ping)
```

Terminal

```
>python client.py
Received pong from
('127.0.0.1', 12345) in
0.00 seconds.
```

Result:

Thus a ping program has been executed successfully.