EX: 13
Date: 25/10/24

Ping Program

Aim:

To implement your own ping program

Algorithm:

UDP server

→ create UDP socket & bind it to a specific addrell & port
→ wait for mellage
→ Print mellage & clients addrell
→ send back Pong to client

UDP client

→ create UDP Socket & set a 2sec timeout
→ if a relponse C"Pong is received - Print relponse & calculate RTT
→ if no relponse within 2 Sec Print request time out
→ Send "Ping" to server

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.recieved(1024)
    Print(f"Received message from {addr}:
                {data.decode()}")
    s.send to(b'Pong', addr)

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

Client.py

```
Import tme
Import socket
def Ping_Server(host='127.0.0.1', Port=12345
    with socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        tay:
            S.Settime(2)
            start = time.time()
            s.send to(b'Ping', (host, Port))
            data, addr = s.recvfrom(1024)
            end = time.time()
            Print(f" Revered {data.decode()} from {addr};
                {end - Start:.2f} second")
            except socket.timeout:
                Print("Request timed ont")

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

Output:

Terminal

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

Received message from
('127.0.0.1', 50051): Ping

terminal

> Python client.py
Received Pong from ('127.0.0.1',
12345) in 0.00 seconds

Result:

Thus a Ping Program had been executed
Successfully