CS23532-COMPUTER NETWORKS-LAB MANUAL
Practical -13

Name: Tharunraj
RegNo:230701362

AIM: - Implement your own ping program

Algorithm:

1. Create a socket (UDP or ICMP) and set a timeout to wait for the reply.
2. Send a small data packet (e.g., "ping") to the target host with a timestamp.
3. Wait for the response; if received, calculate and display the **Round-Trip Time (RTT)**, else report **Request Timed Out**.

Server Algorithm:

1. Create a UDP socket, bind it to a specific IP address and port.
2. Wait to receive data ("Ping") from the client.
3. On receiving data, print the message and send back a reply ("Pong").

Client Algorithm:

1. Create a UDP socket and set a timeout period.
2. Send a "Ping" message to the server and record the start time.
3. Receive the "Pong" reply and calculate the round-trip time, else display "Request timed out."

Program:
sender.py

```
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.settimeout(2)
            start = time.time()
            s.sendto(b'Ping', (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()
```

Receiver.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()
```

Output(server.py):

```
UDP Server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 51234): Ping
```

Output(client.py):

```
Received Pong from ('127.0.0.1', 12345) in 0.01 seconds
```

RESULT:

Thus the program to implement ping has been executed successfully.

Input:-

Output:-