Exp:13
Date: 18/10/24

PING PROGRAM

Aim:
To implement your own ping program

Algorithm:

UDP server:

→ create UDP socket & bind it to a specific address & port
→ wait for message
→ print message & client address
→ send back pong to client

UDP client:

→ Create UDP socket & set a 2 sec timeout
→ Send "ping" to server
→ If a response ("pong") received, print response & Calculate RTT
→ If no response within 2 sec print request timed out

Code

server.py

```
import socket
```

```
def start_server(host='127.0.0.1', port=
                 12345):
with socket.socket(socket.AF_INET,
  Socket.Socket_D 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'pin' addr)
if __name__ == "__main__":
    start_server()
```

Client.py

```
import time
import socket
def ping_server(host='127.0.0.1', port=
    127.... with socket.
socket(socket.AF_INET, Socket.
    SOCK_DGRAM) as :.
try:
    s.settimeout(2)
```

```
start = time.time
s.sendto(b'pin
data, addr =
end = time.t
print(f"Rec
    {addr}
except sock
    print("R
if __name__ ==
    ping_se
```

Output:

Running

=) Python s
UDP serv
on 127.

=) Receiv
('127.

```
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()
```

Output:

Terminal

1) Python server.py
UDP server running
on 127.0.0.1:1234

2) Received message from
('127.0.0.1', 50001): ping

Terminal

> python client.py
Received pong from
('127.0.0.1', 1234)
in 0.00 seconds

Result:

Thus the this program has been executed successfully.

Ex.No:14

Date:22.10.2

Aim:

Implem

sockets

CODE:

from

from

def pa

if ID