Ex.no:13

Practical - 13

Aim:

Implement your own ping problem

Algorithm

Server side (UDP)

1. Create and bind a UDP Socket to a host and port
2. Listen for incoming message in a loop
3. When a message is received echo it back to ~~the~~ Sender
4. Repeat

Client Side (UDP)

1. Create a UDP Socket and set a timeout
2. For each ping
   - Record the start time
   - Send a ping message to server
   - If no response print "Request time out"

Source 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.recvfrom(1024)
 Print(f" Received message from {addr}:
   {dat.adde py}:

     ?..Send to (b' Port', addr)
    if _name_ == '_main_':
    Start-Server()

Client.py
    import socket
    import time
    def ping_server(host='127.0.0.1',
                 port= 12345
  ass:
     try:
      s. Settime out(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:.4f} seconds")
     except socket.timeout
       Print (" request time out")
     if _name_ == "_main_":
       Ping Server()
```

Output

```
UDP Server Running on 127.0.0.1:12345
Received message from ('127. 0.0.1', 54761): Ping
Received message from ('127.0.0.1', 48076): Ping
Received message from ('129.0.0.1', 48706): Ping
```

Result:-

Thus Ping program is implemented and executed successfully.