

Practical - 13
Aim: Implement your own ping program

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}: {data.decode()}")
s.sendto (b'pong', addr)
start = server()
```

client.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")
```

The image shows a Windows 10 desktop environment. A text editor window is open, displaying a Python script. The script defines a UDP server that listens on port 12345 and responds to ping requests. A Command Prompt window is also open, showing the execution of the script. The taskbar at the bottom includes the Start button, a search bar, and several application icons. The system tray shows the date and time as 10:22 AM on 06/11/2024.

```
1 import socket
2 import sys
3
4 def udp_server():
5     server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
6     server_socket.bind(('127.0.0.1', 12345))
7     while True:
8         data, address = server_socket.recvfrom(1024)
9         print(f'Received message from {address}: {data}')
10
11 if __name__ == '__main__':
12     udp_server()
```

Command Prompt

```
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jaya>cd C:\Users\jaya\Desktop\OVCN Lab\Practical-13

C:\Users\jaya\Desktop\OVCN Lab\Practical-13>python server.py
UDP Server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 55798): Ping

C:\Users\jaya\Desktop\OVCN Lab\Practical-13>
```

python 3.11.0

Type here to search

10:22 AM
06/11/2024

```
except socket.timeout:  
    print ("Request timed out")  
    ping - server 1)
```

Output

Result:

Thus the program is successfully executed and output is verified.