**Ex. No.** 03 **Socket Programming (UDP)**

**Date:** 19-08-2025

**Aim:**

To implement and demonstrate communicate between udp client and a udp server using python scoket

**Algorithm:**

1. Import the socket module.
2. Create a socket object using socket.socket(socket.AF_INET, socket.SOCK_DGRAM) for both client and server.
3. For the server:

   → Bind the socket to a specific IP address and port using bind().

   → Wait to receive data using recvfrom().

   → Display the received message.

   → Send a reply back to the client using sendto().
4. For the client:

   → Send a message to the server using sendto().

   → Wait for a response from the server using recvfrom().

   → Display the server's reply.
5. Close both sockets after communication is done.

**********

**UDP Server (udp_server.py)**

```
import socket

# Create UDP socket
server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind to IP and port
server.bind(('localhost', 9999))
```

```
print("UDP Server is ready and waiting for messages...")

# Receive message from client
data, addr = server.recvfrom(1024)
print("Received from client:", data.decode())

# Send reply
server.sendto(b"Hello Client! Message received successfully.", addr)

# Close socket
server.close()
```

**********

**UDP Client (udp_client.py)**

```
import socket

# Create UDP socket
client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Server address
server_address = ('localhost', 9999)

# Send message to server
client.sendto(b"Hello Server! This is UDP Client.", server_address)

# Receive reply from server
data, addr = client.recvfrom(1024)
```

```
print("Server says:", data.decode())

# Close socket

client.close()
```

**********

Sample Output:

Server Side Output:

UDP Server is ready and waiting for messages...

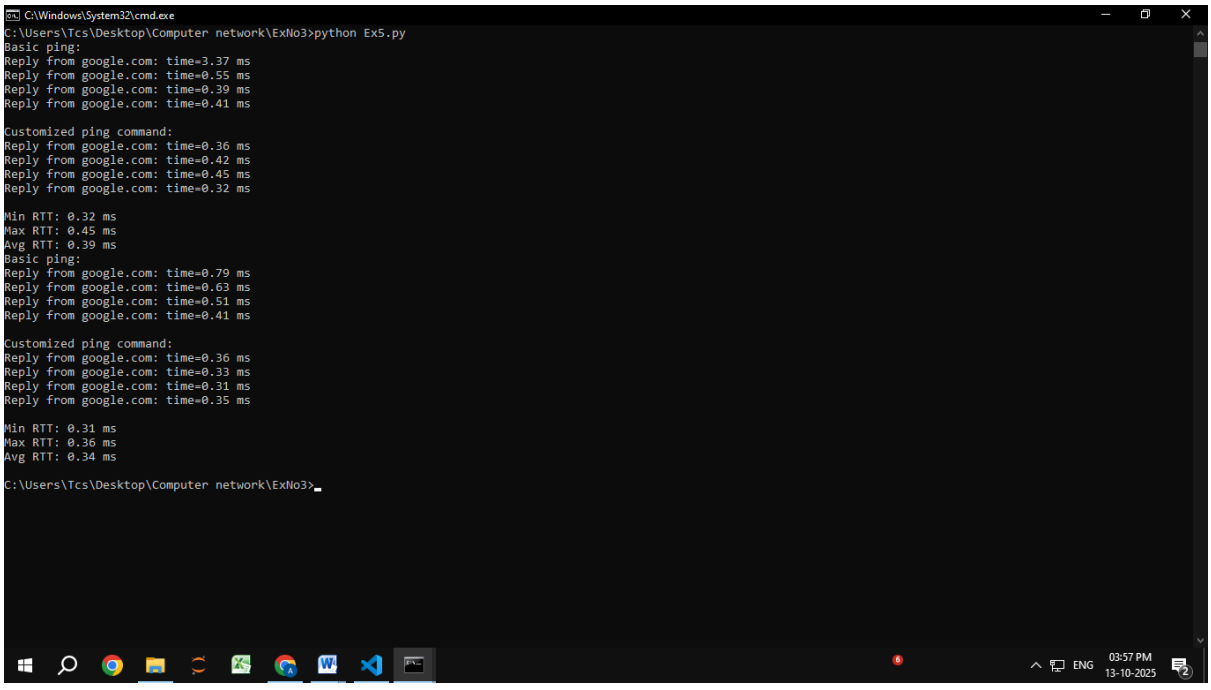
Received from client: Hello Server! This is UDP Client.

Client Side Output:

Server says: Hello Client! Message received successfully.

**********

Output:




**Result:**

Thus, the communication between the UDP client and UDP server was successfully implemented and demonstrated using Python socket programming.