

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 socket

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...

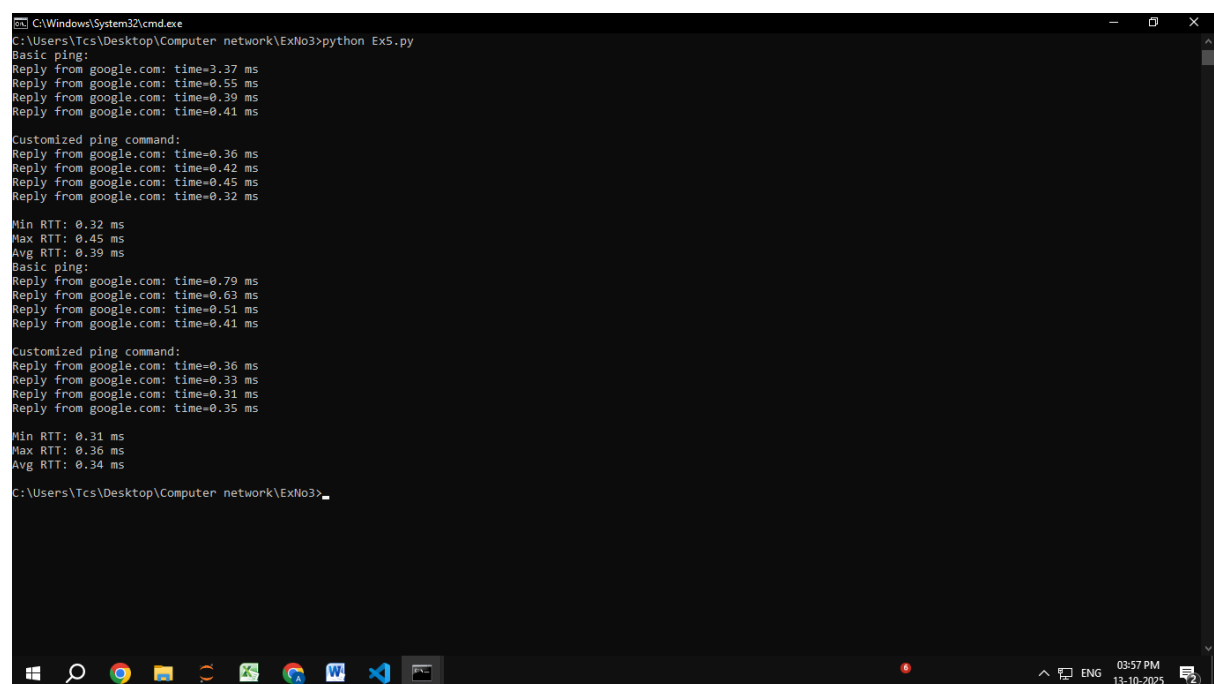
Received from client: Hello Server! This is UDP Client.

Client Side Output:

Server says: Hello Client! Message received successfully.

```
*****
```

Output:



The screenshot shows a Windows command prompt window titled "C:\Windows\System32\cmd.exe". The user has executed the command "C:\Users\Tcs\Desktop\Computer network\ExNo3>python Ex5.py". The output displays two sets of ping results to google.com. The first set shows a "Basic ping:" with four replies and a "Customized ping command:" with four replies. The second set shows another "Basic ping:" with four replies and a "Customized ping command:" with four replies. At the bottom, the user's current directory is shown as "C:\Users\Tcs\Desktop\Computer network\ExNo3>". The taskbar at the bottom includes icons for Windows, search, Chrome, File Explorer, and other applications, with a system clock showing 03:57 PM on 13-10-2025.

```
C:\Windows\System32\cmd.exe
C:\Users\Tcs\Desktop\Computer network\ExNo3>python Ex5.py
Basic ping:
Reply from google.com: time=3.37 ms
Reply from google.com: time=0.55 ms
Reply from google.com: time=0.39 ms
Reply from google.com: time=0.41 ms
Customized ping command:
Reply from google.com: time=0.36 ms
Reply from google.com: time=0.42 ms
Reply from google.com: time=0.45 ms
Reply from google.com: time=0.32 ms
Min RTT: 0.32 ms
Max RTT: 0.45 ms
Avg RTT: 0.39 ms
Basic ping:
Reply from google.com: time=0.79 ms
Reply from google.com: time=0.63 ms
Reply from google.com: time=0.51 ms
Reply from google.com: time=0.41 ms
Customized ping command:
Reply from google.com: time=0.36 ms
Reply from google.com: time=0.33 ms
Reply from google.com: time=0.31 ms
Reply from google.com: time=0.35 ms
Min RTT: 0.31 ms
Max RTT: 0.36 ms
Avg RTT: 0.34 ms
C:\Users\Tcs\Desktop\Computer network\ExNo3>
```

Result:

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