

EXPT: 3 UDP CLIENT-SERVER COMMUNICATION USING SOCKET PROGRAMMING IN PYTHON

Aim :

Develop a simple UDP server and client using Python's socket module to exchange messages.

Algorithm :

1. Server:

- o Create a UDP socket and bind it to an IP and port.
- o Wait for incoming messages.
- o Receive data, print it, and optionally send a reply.

2. Client:

- o Create a UDP socket.
- o Send message to server IP and port.
- o Receive reply from server and display it.

Code :

Server:

```

import socket

sockfd = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
print('UDP Socket Created')

sockfd.bind(('localhost', 55555))
print('Waiting for messages')

while True:

    data, addr = sockfd.recvfrom(1024)
    receivedMsg = data.decode()
    print("Received message from", addr)
    print("Message:", receivedMsg)

    # Send the same message back to client
    sockfd.sendto(data, addr)
    print("Message reply sent to Client!")

```

```
choice = input("Do you want to continue (type y or n): ")  
if choice == 'n':  
    break  
sockfd.close()
```

Client:

```
import socket  
  
clientfd = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
  
server_address = ('localhost', 55555)  
  
name = input("Enter your message: ")  
  
clientfd.sendto(name.encode(), server_address)  
  
data, _ = clientfd.recvfrom(1024)  
  
print("Message Received from Server:", data.decode())  
  
clientfd.close()
```

Output :**Server:**

```
~/udp  
> python server.py  
UDP Socket Created  
Waiting for messages  
Received message from ('127.0.0.1', 43683)  
Message: Hi  
Message reply sent to Client!  
Do you want to continue (type y or n): █
```

Client:

```
~/udp
> python client.py
Enter your message: Hi
Message Received from Server: Hi
```

Result :

The UDP client successfully sent messages to the server, and the server received and replied to each message.