Experiment : 6
Date : 29/05/2021
Author : Bonnie Simon

Client Server Communication using UDP

AIM

Implement Client-Server communication using Socket Programming and UDP as transport layer protocol

THEORY

UDP or User Datagram Protocol is connectionless protocol which is suitable for applications that require efficient communication that doesn't have to worry about packet loss.In communications using UDP, a client program sends a message packet to a destination server wherein the destination server also runs on UDP

- The UDP does not provide guaranteed delivery of message packets. If for some issue in a network if a packet is lost it could be lost forever.
- Since there is no guarantee of assured delivery of messages, UDP is considered an unreliable protocol.
- The underlying mechanisms that implement UDP involve no connection-based communication. There is no streaming of data between a UDP server or and an UDP Client.
- An UDP client can send "n" number of distinct packets to an UDPserver and it could also receive "n" number of distinct packets as replies from the UDP server.

ALGORITHM

Server

1. Start
2. Import socket library
3. Instantiate socket object
4. Bind the localhost address and port to the socket object instance.

5. While True
    a. Listen for data
    b. Print data if data is received
    c. Reply
6. End

Client

1. Start
2. Import socket library
3. Instantiate socket object
4. While True
    a. Read input from user
    b. Send message from user to server using socket instance
    c. Print reply from server
5. End

PROGRAM

Server.py

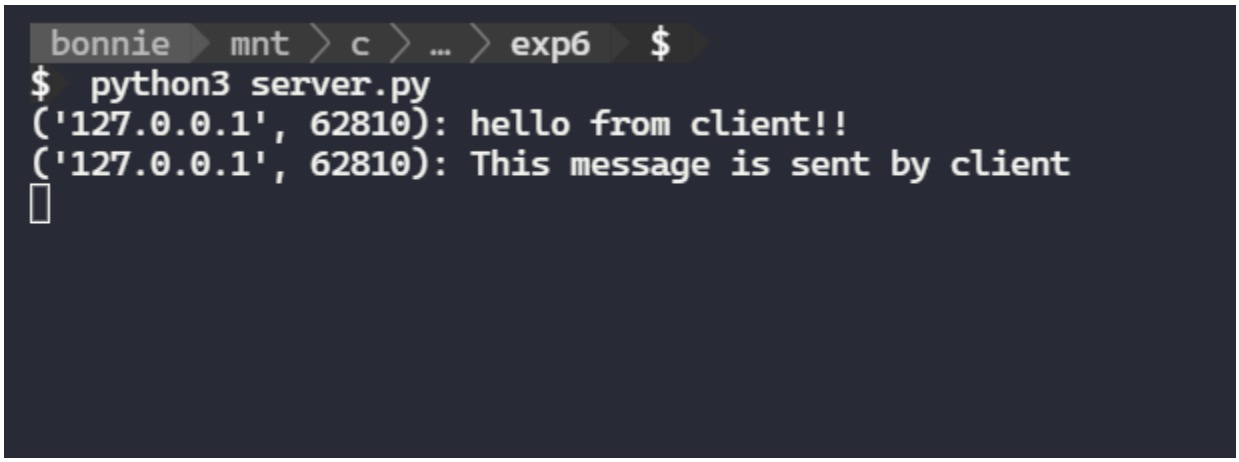
```python
import socket

# set up the socket using local address
socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
socket.bind(("127.0.0.1", 9999))

while 1:

    # get the data sent to us
    data, ip = socket.recvfrom(1024)

    # display
    print("{}: {}".format(ip, data.decode(encoding="utf-8").strip()))

    # echo back
    socket.sendto(data, ip)
```

Client.py

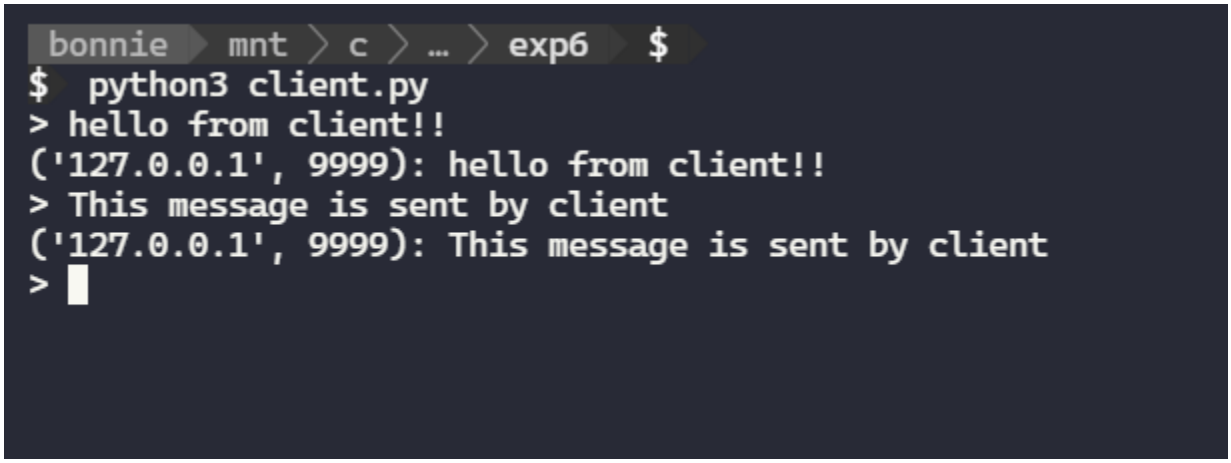
```python
import socket

# create our udp socket
try:
    socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
except socket.error:
    print("Oops, something went wrong connecting the socket")
    exit()

while 1:
    message = input("> ")

    # encode the message
    message = message.encode()

    try:
        # send the message
        socket.sendto(message, ("127.0.0.1", 9999))

        # output the response (if any)
        data, ip = socket.recvfrom(1024)

        print("{}: {}".format(ip, data.decode()))

    except socket.error:
        print("Error! {}".format(socket.error))
        exit()
```

OUTPUT

Server



Client



RESULT

The python program to implement Client-Server communication using Socket Programming and UDP as transport layer protocol has been executed successfully and verified.