Networking Lab Assignment 8

Socket Programming -UDP

Albin Antony

14 March 2019

1 Socket Programming-UDP

1.1 Aim

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

1.2 Theory

1.2.1 UDP

UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (udp) used primarily for establishing low-latency and loss-tolerating connections between applications on the internet. It is a process to process communication. It is unreliable.

1.2.2 Client, Server and Socket

• Server-A server is a software that waits for client requests and serves or processes them accordingly.
• Client- a client is requester of this service. A client program request for some resources to the server and server responds to that request.
• Socket- Socket is the endpoint of a bidirectional communications channel between server and client. Sockets may communicate within a process, between processes on the same machine, or between processes on different machines. For any communication with a remote program, we have to connect through a socket port.

1.3 Algorithm

1.3.1 Server

Algorithm 1 Algorithm for creating a udp server

```
START
Create UDP SOCKET
Bind SOCKET to a PORT
WHILE TRUE:
    RECEIVE message and address(IP,PORT) from CLIENT
    SEND message to CLIENT address
STOP
```

1.3.2 Client

Algorithm 2 Algorithm for creating a udp client

```
START
CREATE UDP SOCKET
SEND message to server at IP and PORT
RECEIVE message and address from server
STOP
```

1.4 Program

1.4.1 Server

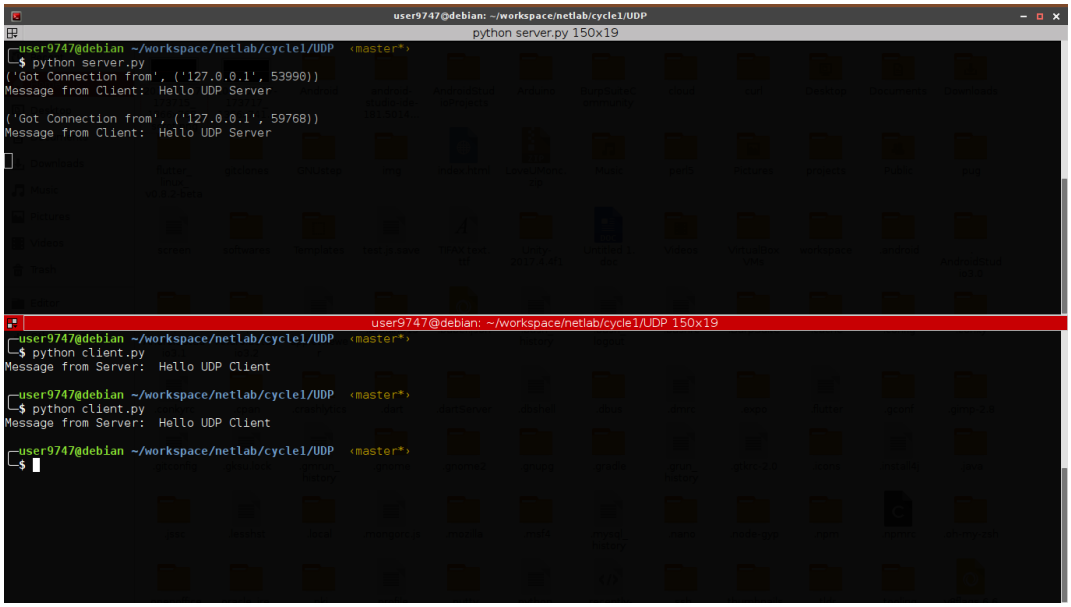
```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

port=8080

s.bind(('',port))


msgFromServer        = "Hello UDP Client\n"

bytesToSend          = str.encode(msgFromServer)

while True:
    msg,addr=s.recvfrom(1024)
    print ('Got Connection from',addr)
    print 'Message from Client: ',msg
    s.sendto(bytesToSend,addr)
```

1.4.2 Client

```
import socket

msgFromClient        = "Hello UDP Server\n"

bytesToSend          = str.encode(msgFromClient)

s= socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

port = 8080
```

```
s.sendto(bytesToSend,('127.0.0.1',port))
msgFromServer=s.recvfrom(1024)
# msg = "Message from Server:  {}".format(msgFromServer[0])
print"Message from Server: ",msgFromServer[0]
s.close()
```

1.5 Output



1.6 Result

Implemented UDP Socket Communication on Python 2.7.13 and executed on Debian 4.9 Kernel 4.9 and outputs were verified.
Server code creates a udp socket using the socket library.Then binds the server to port 8080.In a infinite while loop the server receives message and address from sending clients.Then it sends message to this address.
Client code creates a UDP socket same as above.Then sends a message to the ip and port of the server.It then receives a message from the server.Then displays the message from the server.Then closes the socket.