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

Amrith M

March 6, 2018

1 About

A client/server application model typically is viewed as a remotely located, high powered computing device that stores a large quantity of data with business logic to access them in a network. The user interface is handled by the client software on a relatively cheap machine. This idea is not distinct because any machine serving the request can potentially be called a server. Although the server waits for the client to start a conversation, in some cases the same program may act as both client and server. In that sense, a single machine can act as a network providing the communication between the client and the server program that goes through layers of a TCP/IP protocol stack. A socket is an API provided by the OS to realize the connection.

2 Server.py

```
import socket
import sys

sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
sock.bind(('192.168.1.148',3000))
sock.listen(1)

while True:
        connection,client = sock.accept()
        try:
                print("Client is : ",client)
                while True:
                        buff = connection.recv(16)
                        print("Received Data : ",buff)
                        if buff:
```

```
                                print("Writing Data Back To Client")
                                connection.sendall(buff)
                        else:
                                print("Empty Buffer")
                                break
        finally:
                connection.close()
```

3 Client.py

```
import socket
import sys

sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
sock.connect(('192.168.1.102',3000))

while True:
        try:
                msg = raw_input()
                sock.sendall(msg)
                amount_received = 0
                amount_expected = len(msg)

                while amount_received < amount_expected:
                        buff = sock.recv(16)
                        amount_received += len(buff)
                        print("Packet Sent")
        finally:
                pass
```