

Socket Programming & Porting Linux on (Raspberry/ARM)

Jay Shakti¹, Taniya Das², Rachit Gupta³, Arimardan Singh⁴

*Department of Electronics & Communication Engineering,
IIMT College of Engineering Greater Noida, (India)*

ABSTRACT

The aim of the paper is to introduce sockets, its deployment pertaining to network programming. Sockets play a vital role in client server applications. The client and server can communicate with each other by writing to or reading from these sockets. They were invented in Berkeley as part of the BSD flavor of UNIX operating systems. And they spread like wildfire with the Internet. This paper introduces elements of network programming and concepts involved in creating network applications using sockets. One of the most basic network programming tasks likely to be faced as a java programmer is performing the socket functions/methods because java has been preferred mostly for establishing client server communications using sockets.

INTRODUCTION

In the 1980s, the US government's Advanced Research Projects Agency (ARPA) provided funds to the University of California at Berkeley to implement TCP/IP protocols under the UNIX operating system. During this project, a group of Berkeley researchers developed an application program interface (API) for TCP/IP network communications called the socket interface. The socket interface is an API TCP/IP networks i.e. it defines a variety of software functions or routines for the development of applications for TCP/IP networks. The socket interface designers originally built their interface into the UNIX operating system. However, the other operating systems, environments, such as Microsoft Windows, implement the socket interface as software libraries.

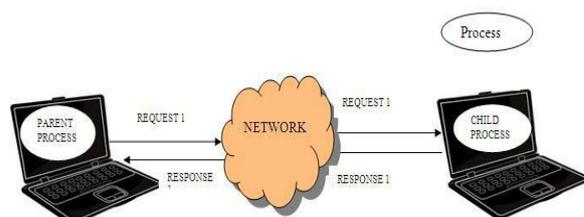


Figure: 1.1 Client-Server communications

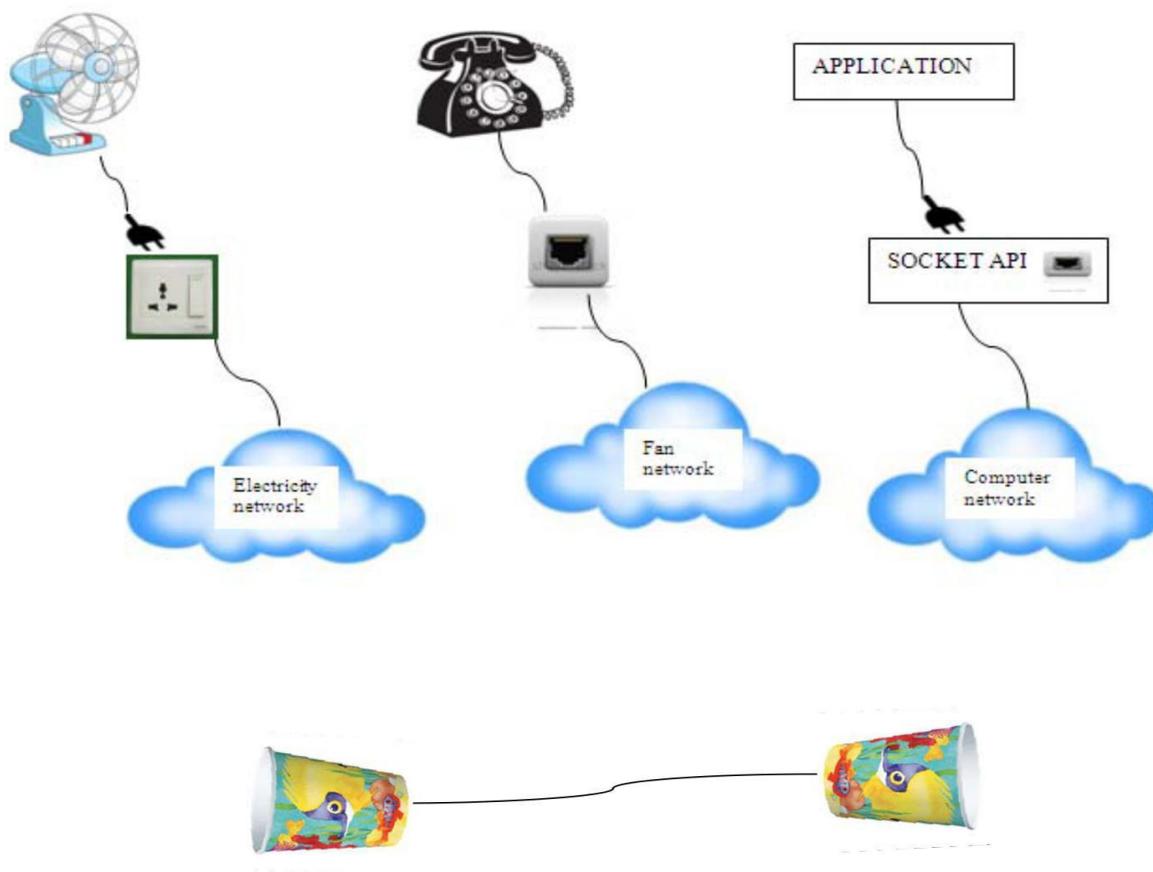
These sockets are the programming interfaces provided by the TCP and UDP protocols for stream and datagram communication respectively of the transport layer which is a part of the TCP/IP stack. When creating a network application, the developer's main task is to write the code for both the client and server programs. The client/server application that is covered here is a proprietary client/server application. A single developer (or development team) creates both the client and server programs, and the developer has complete control over what goes in the code. But because the code does not implement a public-domain protocol, other independent developers will not be able to develop code that interoperates with the application. When developing a proprietary application, the developer must be careful not to use one of the well-known port numbers defined in the RFCs.

II.SOCKETS

This term 'socket' has come from an electricity/phone socket metaphor where sockets acts as interfaces that plug into each other over a network.

Technically, sockets had been defined in many ways.

1. According to Wikipedia, a network socket is an endpoint of an inter-process communication flow



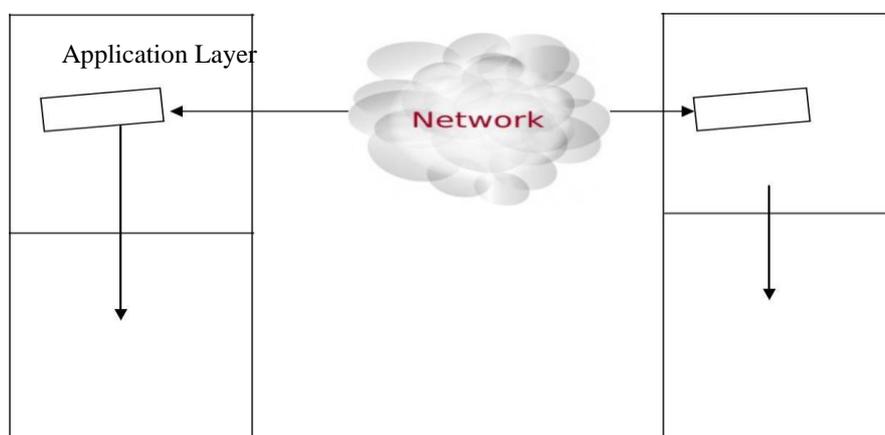
III.NETWORK PROGRAMMING WITH SOCKETS

The Internet has been very popular in the past few years. With its popularity still growing, increased demand for Internet network software has grown as well. One of the greatest advantages to developing Internet software with Java is in its robust networking support built into the core language. The java.net package provides us with classes representing URLs, URL connections and sockets. Combined with the java.io package, we can quite easily write sophisticated platform- independent networking (Internet) applications. Network programming makes use of socket for Interprocess Communication. Due to which Network programming is also termed as socket programming. In Socket programming using Java, BSD style Socket to Interface with TCP/IP services is used. BSD Socket Interface provides facilities for Interprocess Communication. BSD Socket Interface supports different domain, the UNIX Domain, the Internet Domain and the NS Domain. Java Basically supports the Internet Domain to maintain cross platform. In Internet Domain, the BSD Socket Interface is built on the top of either TCP/IP or UDP/IP or the raw Socket. Socket Programming is important to understand how internet based interprocess communication work but not at the level program developed but at a higher level that is compiled to set of Socket Programs. Here sockets can also be termed as network socket or Internet socket since communication between computers is based on Internet protocol.

Socket Programming With TCP:

TCP provides a connection oriented service, since it is based on connections between clients and servers. Connection-oriented means that a connection is established before processes can exchange data. The Transmission Control Protocol is also reliable because when a TCP client sends data to the server, it requires an acknowledgement in return. If an acknowledgement is not received, TCP automatically retransmit the data and waits for a longer period of time.

The processes running on different machines communicate with each other by sending messages into sockets. Each process is analogous to a house and the process's socket is analogous to a door. As shown in Figure 2.1, the socket is



Socket programming over TCP :

Java has provided the facility to create sockets for interprocess communication (IPC). So while programming for sockets in java, one has to make sure to import the java.net package. The java.net package in the Java platform provides a class, Socket that implements the client side connection. And a class ServerSocket that implements the server side connection. The Server Socket on the server performs the methods ‘bind’ which is to fix to a certain port no. and IP address, ‘listen’ to wait for incoming requests on the port and ‘accept’ for acceptance of connection from the client respectively. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint (i.e. socket) set to the name of the machine and port of the client. So the client initiates a three way handshake with the server and creates a TCP connection with the server. The client and server can now communicate by writing to or reading from their sockets. And when communication is done between the client and the server, the close method is called from both client and the server for closing the connection as shown in figure below:

The list is a summary of functions or methods provided by the Berkeley sockets API library:socket() creates a new socket of a certain socket type, identified by an integer number, and allocates system resources to it.listen() is used on the server side, and causes a bound TCP socket to enter listening state. connect() is used on the client side, and assigns a free local port number to a socket. In case of a TCP socket, it causes an attempt to establish a new TCP connection.accept() is used on the server side. It accepts a received incoming attempt to create a new TCP connection from the remote client, and creates a new socket associated with the socket address pair of this connection.send() and rcv(), or write() and read(), or sendto() and rcvfrom(), are used for sending and receiving data to/from a remote socket.close() causes the system to release resources allocated to a socket. In case of TCP, the connection is terminated.

Socket programming over UDP:

UDP is a connection-less, datagram protocol. The client does not establish a connection with the server like in case of TCP. Instead, the client just sends a datagram to the server using the sendto function which requires the address of the destination as a parameter.

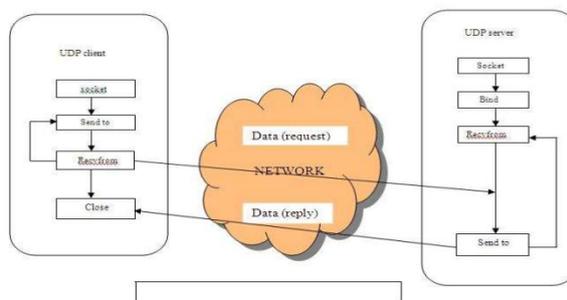


Figure: 2.4 UDP client - server

IV.CONCLUSION

This paper describes the details about sockets, ports, socket programming over TCP and a little bit of UDP. Network programming makes use of socket for interprocess communication between hosts where sockets act as the endpoint of the interprocess communication. Here sockets can also be termed as network socket or Internet socket since communication between computers is based on Internet protocol. So Network programming is also Socket Programming. The paper also describes about socket programming in java over TCP. Because java has been preferred more than any other language for establishing connections between clients and servers using sockets. Socket programming in java is easy.

REFERENCES

- [1.] Emblogic India Pvt. Ltd Noida.
- [2.] James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach featuring the Internet".
- [3.] Joseph M. Dibella , "Socket Programming with Java"
- [4.] http://www.tutorialspoint.com/java/java_networking.htm
- [5.] The Java Tutorials, "Lesson: All about Sockets".
- [6.] RajkumarBuyya, "Socket Programming".
- [7.] http://en.wikipedia.org/wiki/Berkeley_sockets.