Communication between Android (Client) and Linux (Server) using TCP Socket

[1]Swapnil Krishnakumar Padamwar, [2]Alice N. Cheeran, [3]Shweta Yadav
[1]Student of M. Tech (Electronics), [2]Professor, [3]R&D Engineer
[1, 2] Department of Electrical Engineering, VJTI, Mumbai, Maharashtra, India [3]A3RMT Pvt. Ltd.

***Abstract*— TCP/IP protocol is an industry standard model used to deal with networking problems. It is an adaptable customer server design that enables customers to be included without disturbing current administration services. In this paper, client-server model is presented. Android (client) and LINUX (server) uses socket programming to bind them in a network and TCP/IP for communication. Communication can be text communication from server to client and vice-versa or data communication from server to client. For wireless connection, hotspot of client (android device) is used, through which server connects. The client- server model is used for a medical device, where a Client sends request to Linux processor for data acquisition from sensors. The acquired data from sensor (like ECG signals) is processed and sent to client. Data can be a single image or multiple images, text files, etc. Since this model uses simple hotspot of client device for connection and not external gateways or routers, so hardware is reduced.**

***Keywords* – TCP/IP protocol, Client-Server model, Wireless communication, Android, Linux, Socket programming.**

I. INTRODUCTION

Ethernet technology obtained fast development because of its low price and high degree of flexibility. It is not only widely used in the business office area, but also used in a dominant position of the upper network communication market in commercial computer network communications and industrial control systems. Its application in embedded devices is becoming wider and wider and its market share is getting higher and higher. Ethernet data transmission protocols include two main types: one is TCP/IP Protocol, the other is UDP Protocol. TCP / IP provides reliable data transmission in the IP environment, efficient flow control, full-duplex operation, multiplexed services. TCP/IP is connection-oriented, used to transfer large amount of data, with high reliability requirements of the application; UDP is connectionless-oriented, unreliable, used to transmit small amounts of data (packet mode) [1-4]. Therefore, TCP / IP protocol is used in proposed model.

Only a limited number of Ethernet protocols are suitable for integration in real-time interactive applications. Most of these protocols are not designed for real-time and reliability, but reachability is good for all of them since they all provide one or more ways to traverse through firewalls. The messages in these protocols are quite large because they are text-based, especially SIMPLE and XMPP. The request based mechanisms like web-services require higher bandwidth and processing time and double that with the required return messages [5].

Socket: A socket is a termination of a two-way communication link between two different programs running on same network. An endpoint is defined as a combination of an IP address and a port number. TCP layer identifies the application where data is to be sent using the port number with which socket is bound.

The purpose of the proposed scheme is to build a client-server model. This model is used in healthcare systems, where processor and sensors are controlled from mobile devices such as android phones, tablets, etc. The model thus developed, with some modifications, can also be used in general purpose applications, where data/file exchange between Linux and android is a primary goal.

The overview of the paper is as follows: Section II represent proposed scheme, section III explain the working of proposed-model with the help of flow charts, section IV provides the outcomes of the model and section V concludes the proposed scheme.

II. PROPOSED SCHEME

The purpose of this scheme is to establish a communication between server (on Linux system) and client (on Android device). Here the proposed scheme implements a client-server model. Since TCP provides reliable data transmission, efficient flow control and full-duplex operation, it is used for transferring large amounts of data. Due to high reliability requirements of the application, this proposed model uses TCP for message or data transmission. Fig 1 and Fig 2 shows the block diagram of proposed scheme. Fig 1 shows block diagram for client-server text communication and Fig 2 shows block diagram for client-server data communication.

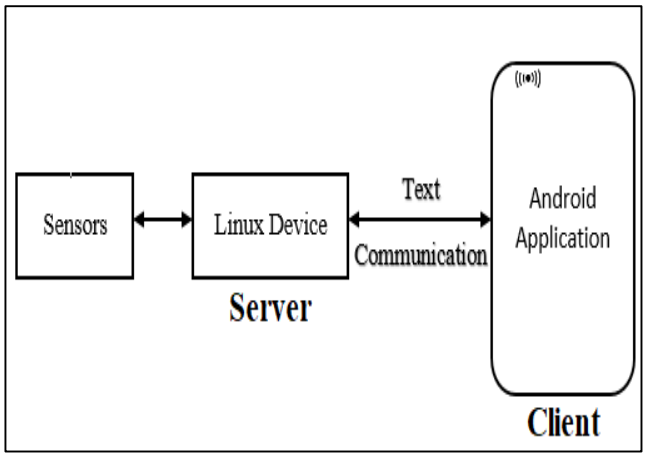

Fig1: Client-Server Text Communication

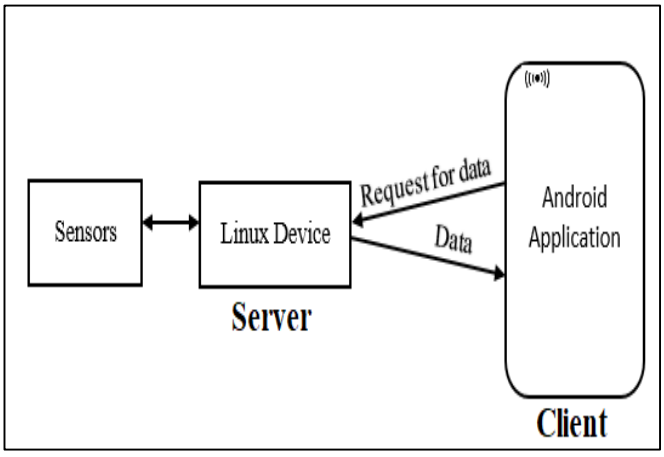

Fig2: Client-Server Data Communication

In proposed scheme, the client-side have a user friendly GUI. The GUI is built using Android Studio, an IDE by Google for Android OS, server runs on Terminal of Linux system. The message is sent or received in the form of bytes. The large data is first compressed and then sent to client, in the form bytes. At the client side, data is received and stored in device memory. As the data is in compressed from, it is first uncompressed and then stored in device memory of android device. On start of android application, hotspot also turns on and server is allowed to connect to it. After the connection is established, socket program runs at server-side that binds server to its own IP and server act as 'Host'. Now, the server is ready to accept the request from client for connection. Client sends request to server and after successful connection, communication between client and server starts.

Features:

- Text Communication: Server and Client communicate with each other via a text message.
- Data Communication: Server sends data to client on its request.

III. METHODOLOGY

Both client and server are programmed in two different programming languages, that are Java and Python respectively, but both uses the concept of socket programming to build up the connection.

1. *Server-side Program Flow:*

Server program is actualized on Ubuntu, using Python. Ubuntu is a free and open-source Linux dispersion dependent on Debian. Python stresses on code intelligibility. Python includes a dynamic kind of framework. It bolsters different programming ideal models including object-arranged, basic, utilitarian and procedural, and has an extensive and complete standard library.

Fig 3 shows a flow chart for server side program. Initially server connects to client's Hot-Spot. The server, using socket function and bind function, server binds itself to device IP address, and acts as a 'host'. Host uses listen and accept functions to listen to client's request and to bind a connection with the client. After successful connection, receive and send functions are used for data/text transferring. Once the data/text transfer is over, close socket function is used to end socket connection and to release the resource allocated for socket. After the execution of close socket function, server again uses listen function to listen to another request from client, and the above process is repeated again.

2. *Client-side Program Flow:*

Client program is actualized on Android OS using Android Studio. Android Studio is IDE for Google's Android OS. It is designed for android development.

Fig 4 shows a flow chart for client side program. As soon as android application starts, Hot-Spot is created. On clicking the button, a new thread starts, having network related functions. Thread uses socket function, IP address and port of host, and sends request to server. After the connection establishment, send or receive function is used data/text transfer. In the end the close socket function is used to end current communication. For new connection, client needs to send another request to server and this process continues.

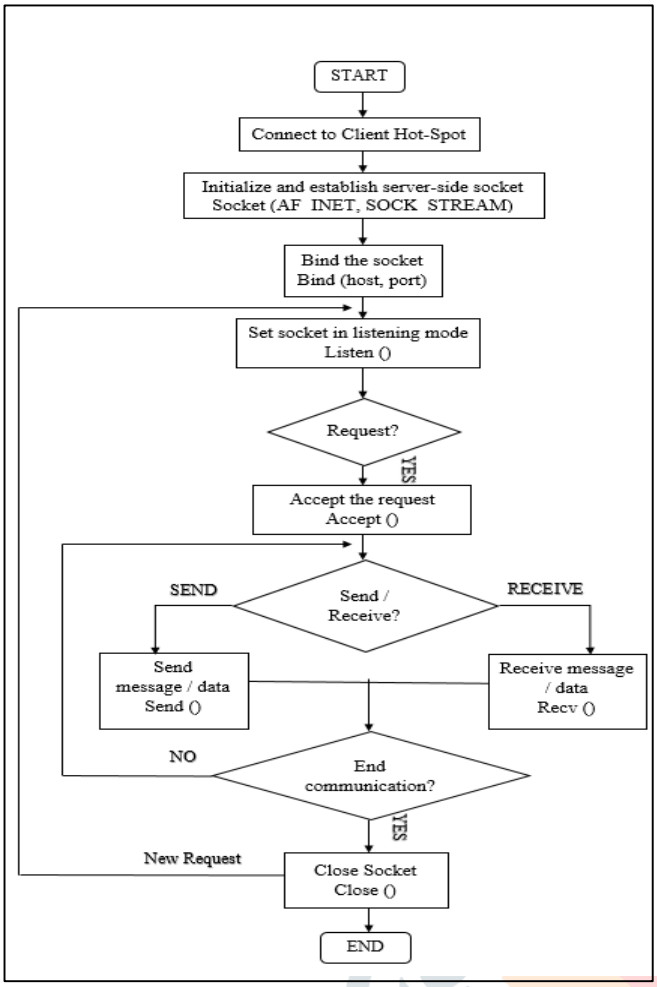

Fig3: Server-side Program Flow

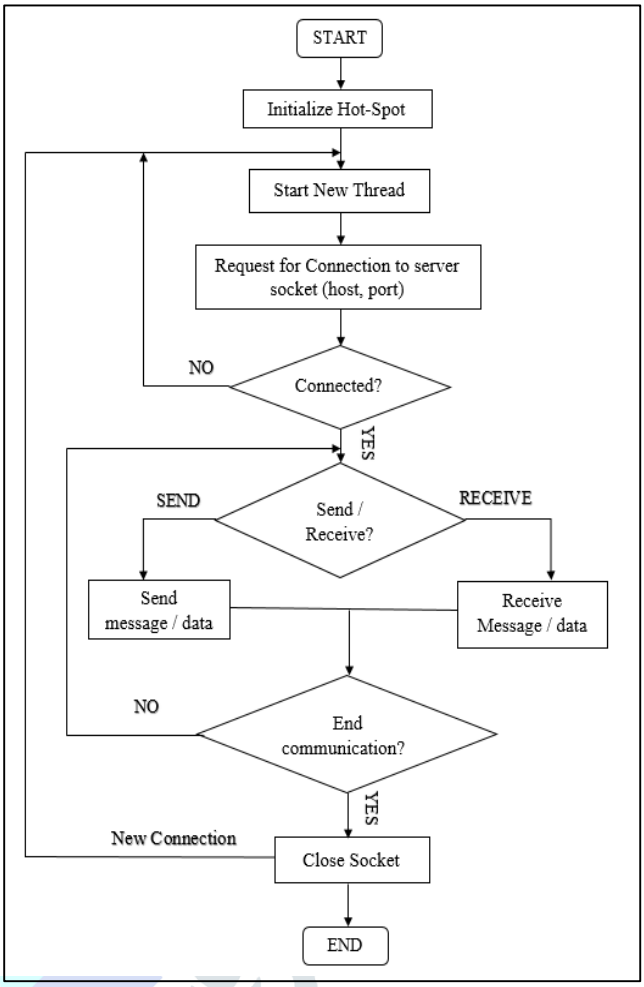

Fig4: Client-side Program Flow

IV. RESULTS

Fig 5 shows, server is bound to its IP address and it is ready to accept the client's request and Fig 6 shows, IP address and port of connected client. The screenshots ensures the connection establishment between client and server.

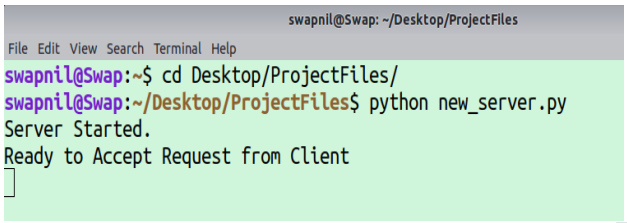

Fig 5: Server is ready to accept client's request

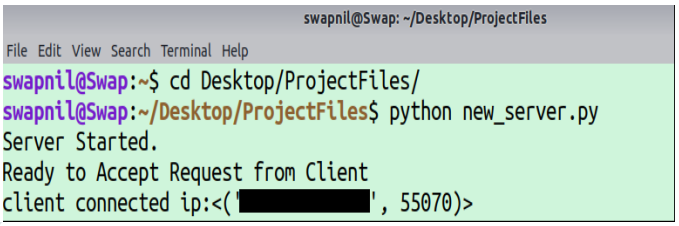

Fig 6: Client is connected to Server

Fig 7 and Fig 8 are screenshot taken at Client side. Fig 7 shows client GUI with button. Button on GUI is used to request server. Fig 8 shows a message on GUI, this message appears on GUI when button is clicked.

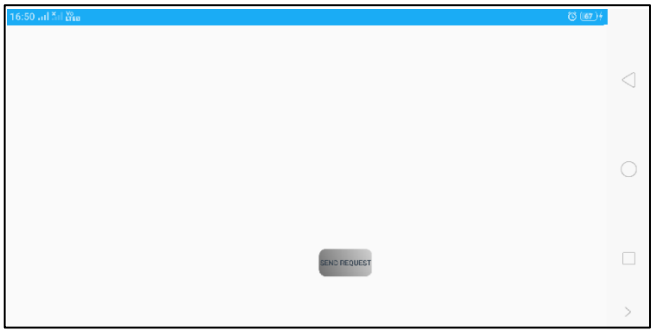

Fig 7: Client-side GUI

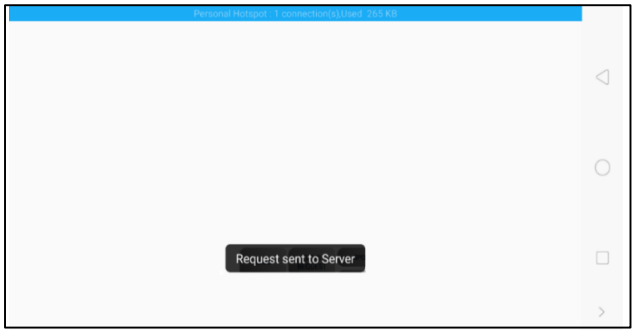

Fig 8: Client send request to Server

Fig 9 is a snapshot taken at server side, displaying the client's message. Fig 9 and Fig 10 are screenshots of client device. Fig 10 shows, 12 lead ECG images displayed on GUI and Fig 11 shows, ECG images are stored in storage directory. Fig 9, Fig 10 and Fig 11 shows that the data sent from server is received and stored by the client, and stored data is displayed on GUI.

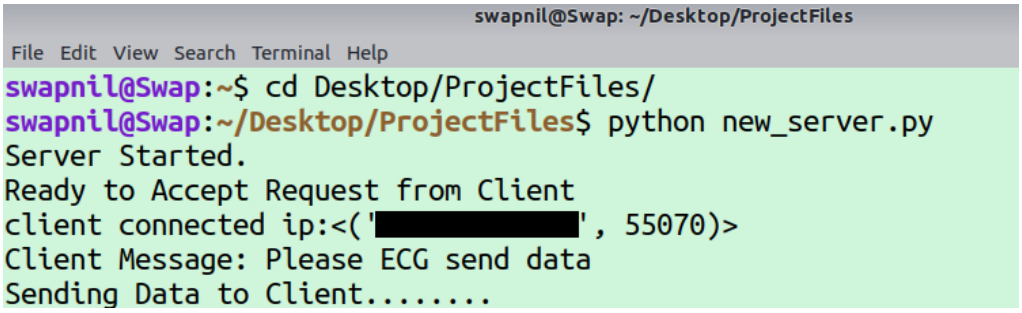

Fig 9: Client message and data sending to Client

Fig 10: Data (12 lead ECG images) received at Client and displayed on GUI

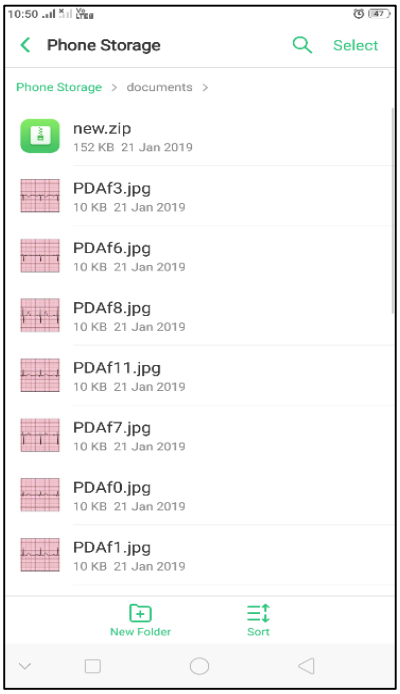

Fig 11: Data stored in Client's android device memory

V. CONCLUSION

The proposed scheme which is based on TCP/IP protocol and socket connection is used to implement a typical client/server model. Python socket programming technique is used at server-side and Java socket programming technique is used at client-side to design a message or data transmission system. The connection is established using client's android device Hot-Spot, which reduces the external hardware like routers and hence this model reduces the cost of communication. The results demonstrates the successful transmission of message and data and along with the successful transmission, the data is stored in client's android device memory. This proposed scheme solves the problem of message or data transmission between the Linux and Android device (or android application).

VI. REFERENCES

[1] D'Ambrosia, J. "The Next Generation of Ethernet". Communications Magazine, IEEE, Vol. 46, No. 1, pp. 8-15, 2008.

[2] Sommer J, Gunreben S, Feller F, Kohn M, Mifdaoui A, Sass D, Scharf J. "Ethernet – A Survey on its Fields of Application". Communications Surveys & Tutorials, IEEE, Vol. 12, No. 2, pp. 263-284, 2010.

[3] Nan Xie, Haibo Zhang, Weimin Chen, Yan Ma, Jinghua Tian. "Research and Design of Industrial Ethernet Intelligent Gateway Based on ARM", The 2008 International Conference on Embedded Software and Systems Symposia, Chengdu, Sichuan, China: ICESS, pp.324-327, 2008.

[4] Baofeng Zhang, Qing Xia and Fangfang Han, "Multi-point-to-point Image Transmission System Based on TCP/IP Protocol", Fifth International Conference on Instrumentation and Measurement, Computer, Communication and control (IMCCC), pp. 1621 – 1624, Sept. 2015.

[5] Zhilong Yang, Yong Wang, Yongquan Yang, Zhiqiang Wei, "Research and Design of a Real-time Interactive Application Development Model Based on the Android Platform", Sixth International Symposium on Computational Intelligence and Design, 2013.

[6] Ming Xue, Changjun Zhu, "The Socket Programming and Software Design for Communication Based on Client/Server", Pacific-Asia Conference on Circuits, Communications and Systems, pp. 775 – 777, May 2009.

[7] Zhaojuan Yue, Xiaodan Zhang, Y ongmao Ren, Jun Li, Qianli Zhong, "The Performance Evaluation and Comparison of TCP-based High-speed Transport Protocols", IEEE International Conference on Computer Science and Automation Engineering, pp 509 – 512, June 2012.

[8] Bo SHANG, Chengdong WU, Tingting MENG, Chengxi GAO, Yunzhou ZHANG, "A Data/Image Transmission Device Based on TCP/IP Protocol", 8th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1-6, Sept. 2012

[9] Min Xiao, Yu-Han Liu, Qian Hu, "Design and Implementation of Socket-based-Network Connections Smart Home System", Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), pp. 756 – 760, July 2016.