IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 2, June 2022

Multi Chat Application based on Qt-X11 G11 ARM Linux

Shrawankumar Udayraj Patel[1] and Dr. Kirti Muley[2]

Student, Bharati Vidyapeeth, Navi Mumbai, Maharashtra, India [1]

Assistant Professor, Bharati Vidyapeeth, Navi Mumbai, Maharashtra, India[2]

Abstract: *Today's world is internet connectivity, everyone is connected to from one to another either by means of social site like Facebook, twitter or LinkedIn for promotion or for searching jobs. But when it comes for office environment intranet communication is very crucial. Organization workings flow depends on employees' ability to quickly ask for and receive information. Instant messaging allows us to communicate with others in real time, with no delays. Messaging system helps to increase team productivity and also cut the inefficient overhead and operating expenditures. This paper is for, the Qt Framework setup with ARM complier, development of software, recompiled in kernel and migration from X86 to ARM. X86 and ARM are the two major Processors hogging the CPU market, ARM Processors deployed in mobile Devices where size, power consumption, speed matters. The ARM processor is the right choice for IOT application, because the embedded system needs to be connected to all equipment, and it should be a little sized.*

Keywords: QT Creator Application, Network Application, Server-client Technology, P2P Technology, Socket Programming.

I. INTRODUCTION

People enjoy using instant messaging. Messaging Technology is so vastly became a part of everyday life that most of the employees will feel completely at ease messaging each other at work. Start-ups and small or mid-sized companies need to constantly reduce small business costs just to survive. In many instances, instant messaging can help to trim inefficient overhead and operating expenditures. This paper is about, GCC and G++ compiler selection, software development with Qt framework, Server-client technology, Network and Socket programming, then transfer of executable files from X86 Linux to ARM system and run executable to that System [2].

This paper is for to establish a connection between GL-11 visionTek POT which is based on ARM Linux kernel System using QT Framework Network Protocol. The other is to develop a Chat application using the Client-Server architecture. This architecture depends on socket programming based on Qt network classes like QTcpSocket, QUdpSocket and QTcpServer [4]. Develop the messaging and file transferring system for private network whether it is small office or large organization or school or colleges. To Enable Easy and Fast way of Communication for Smooth functioning of Organization or Different Department within organization or within groups. Ensure the transfer of files whether it is small or large files without any restriction or internet connectivity [2].

Private Institutions aim at having separate messaging system to communicate with people and to share resources securely to them. Sharing resources through other public applications are not much reliable and considered insecure and exposed to security threats. Public applications are easily affected by network threats, etc. This application extends one-way messaging to multi way communication.

II. FEATURES IMPLEMENTED

2.1 Messaging

Basic and primary requirement for deploying the system is the network connection. The Chat application must allow both sending and receiving process in simultaneous way and this is achieved with QTcpSocket, QUdpSocket and QThread Class in this Application [4].

DOI: 10.48175/IJARSCT-5492

2.2 Group Chat Room

Next important feature in this application is group chat. It allows people to exchange the information and share the files within group. Message will be sent to all the users in group chat room along with the domain name and the name of the user who has sent the message. All Users who are available in the chat room will send and receive the message.

A. 2.3 Private Chat

Another feature of this application is user able to communicate privately. Peer Manager broadcast every user whose is connected to local network within organization. By selecting the user name that is broadcast IP highlighted on, they are able to communicate.

III. ARCHITECTURE

This application has been implemented based on server-client model using Qt framework network class.

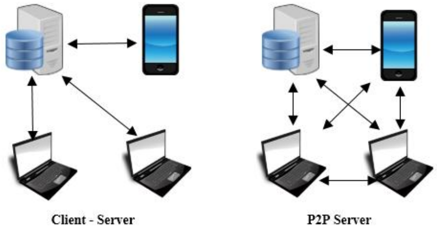


Fig.1. Client Server and P2P Architecture resolution

3.1 Server-Client

Server-client describe the relationship between two or more computer programs in one, the client program makes a service request to server, and the server explains communication between any end nodes and a shared resource (e.g. Fig. 1). In short Server provides the services and the other one that is client consuming those services [5].

3.2 Title P2P Technology

P2P Technology is network model in which all computers can function as clients or servers as needed and there's decentralized control over network resources. Users control their own shared resources and might have several sets of ID to access resources. All computers in the network have equal capabilities to use the resources (data and files) available on the network. Every node is works as a client and as well as acts as Server as shown in Fig. 1 [8].

3.3 TCP Socket

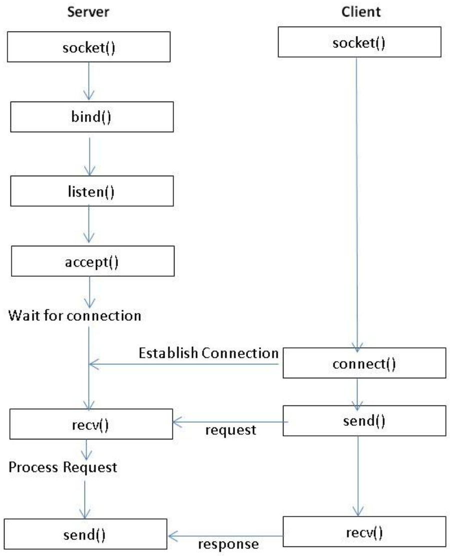


Fig.2. Block Diagram of TCP Socket

Description of TCP Socket, as shown in Fig. 2

Depending on the platform client and server sockets may be implemented differently.

1. Server Socket is created at beginning for new connection which is then binds the socket to local address with IP and Port [9].
2. After that Socket in particular host begins to listen in the particular port. Then server is ready to accept the connection request from the client through a particular port [9].
3. The Socket is instantiated in client side to connect to the server. . A new Server Thread using socket is created to accept all the requests from multiple clients [9].
4. After accepting the request both read and write operation occurs simultaneously, clients who request the server can communicate with each other and share resources [9].
5. Once the communication finished the socket is closed both in the client and server side [9].

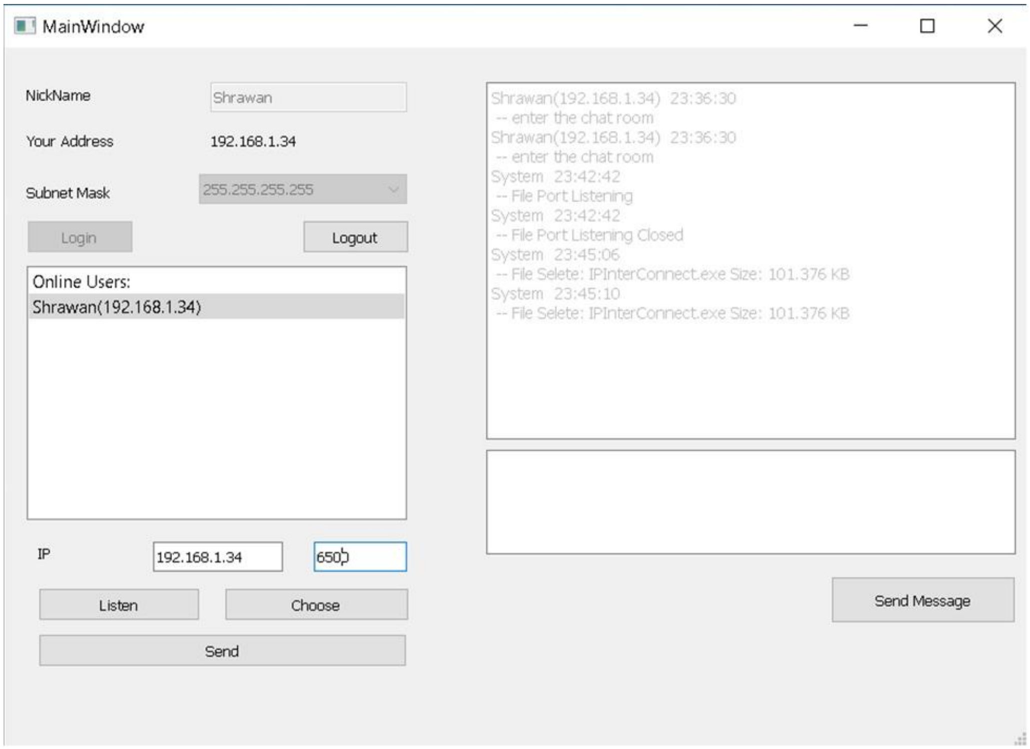
3.4 Qt Framework with QTcpsocket and QTcpserver

- QT framework there is QTcpSocket and QTcpServer Class to establish a connection between Server and Client. The QTcpSocket class provides an interface for TCP and The QTcpServer class provides a TCP-based server. QTcpServer classes accept incoming TCP connections. For Transferring data need to specify the port or have QTcpServer pick one automatically and remote host address [3].
- QTcpServer::listen() function makes the server to listen for incoming connection. The QTcpServer::newConnection() signal which is emitted each time a client connects to the server [3].
- The IP address and port of the peer are available through QTcpSocket::peerAddress() and QTcpSocket::peerPort() after the connection has been established [4].
- In slot, by calling QTcpServer::nextPendingConnection() which accept the pending connection and use the returned QTcpSocket to communicate with the client [4].
- QTcpServer::serverError() function return the type of error and serverAddress() and serverPort() return the address and port on which the server is listening [4].
- QTcpServer::close() makes the server no longer listen for incoming connections [3].

IV. RESULT AND WORKING OF APPLICATIONS

- Start QT Application, Select file menu to create new project, give name to project, then on create build configuration select QT version (QT –X11-ARM Target) then finish [1].
- On Toolbar option select Toolchain as gcc-arm-linux-gnueabi compiler and g++-arm-linux-gnueabi [1].
- In Project pro file add as QT += core gui network then click on build to add network library on file, once the development completed click on build all project [1].
- Open Terminal to transfer build file: type ssh IP_address (linux Command) of the device, once the device is connected make a folder where need to copy the build file using mkdir Filename command. Open new terminal to copying file use scp p 22 build_file_name root@destination_IP: /fileLocation/ command [10].
- In this UDP protocol is used for text messaging while for transferring file TCP protocol as it’s a reliable protocol. User need to add name first then press on connect button to start texting.
- From ListWidget user able to see online connection. On clicking Listen Button make TcpServer socket connection and choosing file user able to send file from send button, as shown in Fig.3

DOI: 10.48175/IJARSCT-5492



NickName: Shrawan
Your Address: 192.168.1.34
Subnet Mask: 255.255.255.255

Online Users:
Shrawan(192.168.1.34)

IP: 192.168.1.34 6500

```
Shrawan(192.168.1.34)  23:36:30
 -- enter the chat room
Shrawan(192.168.1.34)  23:36:30
 -- enter the chat room
System  23:42:42
 -- File Port Listening
System  23:42:42
 -- File Port Listening Closed
System  23:45:06
 -- File Selete: IPInterConnect.exe Size: 101.376 KB
System  23:45:10
 -- File Selete: IPInterConnect.exe Size: 101.376 KB
```

Fig. 3 Server Application

V. FUTURE ENHANCEMENTS

Further enhancements would be involved in the area of saving history of chat, making a log file for tracing activity, large size transfer even folder transfer, encryption of messaging for enhancing security and some additional features that are required in the competing world. Other work is involving in implementation of the system in private networks.

VI. CONCLUSION

Server-Client Application provides a better and efficient system for messaging in private network. It is developed with QT Framework network classes and P2P technologies in a way to provide a reliable system. Main advantages of the system are instant messaging on local and public network connectivity, adding security, group chat, etc. This application can find better need in the market for most of the organizations aim at having private applications for them. Additional features will also be included in the system based on the environment need.

REFERENCES

[1]. Mastering Qt5 second Edition -- by Guillaume Lazar and Robin Penea.
[2]. https://www.scientific.net/AMR.926-930.1906.
[3]. https://docs.huihoo.com/qt/4.7/qtcpserver.html
[4]. https://www.bogotobogo.com/Qt/Qt5_QTcpServer_Client_Server.php.
[5]. https://www.geeksforgeeks.org/simple-client-server-application-in-c.
[6]. https://www.educba.com/arm-vs-x86/.
[7]. https://www.acmesystems.it/arm9_toolchain.
[8]. https://www.geeksforgeeks.org/what-is-p2ppeer-to-peer-process/
[9]. https://www.geeksforgeeks.org/socket-programming-cc/
[10]. https://www.geeksforgeeks.org/scp-command-in-linux-with-examples/

DOI: 10.48175/IJARSCT-5492