Accessing a Remote Desktop of a Client Using Remote Frame Buffer Protocol through Virtual LAN Cable

[1]M.Maheswari,[2]Anindita Bora,[3]Apurv Saxena
[1]Assistant Professor,[2] Student,[3] Student
[123]Department of Software Engineering,
[1]SRM University, Chennai, India

***Abstract*—Remote Desktop Sharing is a client application that allows us to view or access the desktop session on another machine that is running on a compatible server. In this, Client Network is created after authentication where a server can see what is running on the remote host (client). Once the Client is connected to the Server the server sends images of its screen to the client periodically. Application hosted on the client side illustrates these image feed into it integrated environment that allows client to control the remote desktop by continuously communicating mouse and keyboard activities to remote host. Hence, allowing client to perform various types of computation on the server side.**

***IndexTerms*—Remote Frame Buffer Protocol, robot class, file transfer, screen sharing, video streaming.**

I. INTRODUCTION

Remote Desktop Sharing is a client application that allows users to view and control the desktop session on remote device having compatible environment. The integration approach of our application and protocol used are as follows.

Remote Frame Buffer protocol is widely used to establish Remote Desktop Connections. Devices that implements high level of a Remote Frame Buffer protocol are known as Virtual Network Computer (VNC). It is a simple protocol for remote access to graphical user interfaces of a remote device. It works at the frame-buffer level, which roughly corresponds to the rendered screen image, which means that it can be applied to all windowing systems (including X11, Mac OS and Microsoft Windows). Remote Frame Buffer applications exist for many platforms, and can often be free re-distributed. It takes a reasonable amount of network traffic to send an image of the frame buffer, so it works best over high bandwidth links, such as local area networks.

To overcome the issue of the limited bandwidth, we decided to transfer only rendered pixels in continuous frames hence decreasing amount of data to be transferred significantly. To decrease the size of data even more, certain compression and hashing techniques were used. Hence, the application could be used for low bandwidth networks with minimal latency.

The compression and hashing techniques in our application were used to make sessions between two devices secure and safe. Also we used handshaking and password authentication while establishing session to enhance the session security especially on the client.

FTP protocol is also used by our application to perform file transfer between server and client. We have implemented drag and drop feature to make file transfer even simpler.

Application Integration-To establish connection between server and client two approaches could be used; server name entry and invitation from the server. In server name entry user is asked to enter the host name or IP address of the server along with port to establish a connection on a specific port. Another method known as invitation uses a handshaking protocol where server would connect with client first (invite client). Once the client accepts the invitation the connection will be established between the server and client.

II. RELATED WORK

An approach to expand the various features of several Remote Desktop sharing application that are built with limited features[2]such as sharing screens or viewing operations on server or client side for tutorial purpose. Some of the applications such as team viewer, telnet, skype uses similar remote buffer protocol to share the screen and use various input devices such as web came, keyboard, mouse, microphone and many more. Some real-time remote desktop sharing software[2], have recently been getting popular.

We can design a presentation system that would help lecturers deliver their desktop screen images to students via network in classes at their University. The system was based on the following design policy: 1. Lecturers should be able to use any application software on their PC. 2. Student PCs should not require any client software. VNC [6] is a remote desktop sharing software working on multiple operation systems. Users do not have to install any special software in using the VNC Java Applet. VNC is designed to only support communication between two computers. But the capability of transmitting data from one to more than 100 remote users is not possible . Because of this, VNC cannot be applied to this case. Further to this problem, VNC uses a dedicated network protocol that differs from HTTP. This makes it difficult for VNC connections to pass through corporate firewalls

III. Proposed Work

It is very important to ensure the compatible configuration of firewall on both devices. While establishing connection user will be authenticated and a secure session will be created from where client would be able to control and perform various operations on the server side. To allow client to perform various operations it is very important to replicate mouse and keyboard activities on client side to the server side. To integrate the mouse and keyboard activity between server and client, API is used to access the keyboard and mouse activity on the client side. With the help of API, activity data will be collected which then will be passed over the network to server side. These data would be used as input to another API, which will replicate the activity as per the data. Hence, allowing clients to control the movements on the server side. To implement these operations traditional windowing system would be used allowing user to access the remote host in either full screen or windowed screen mode, which can be configured to desirable size. The session would be terminated once user clicks logout button. While logging out if user was in full screen mode then he would be transferred to windowed mode once the user has been logged out.
Minimum of 1 GB computation power is required to use the application effectively. Also it is very important Java Development kit is installed on server as well as the client side. Last but the least, firewall should be configured in an application friendly way on both of the machine

Functionality-To make session secure and enhance the user experience, the application users will be allowed to approve or reject a request to remote connection. Once the connection is established user would be able to control the interaction level, visibility of the desktop and to terminate a connection with other device. User will also be able to control the windowing system of the application as well as transfer the files and view files between the devices. All the functionality mentioned above will be compatible with low bandwidth devices with minimal latency and maximum security between the devices. Also the application is platform independent has is runnable between inter-operating systems.

Architecture of Application-Functionality mentioned above is implemented using six different modules. They are listed below.

User Authentication Module-Main responsibility of this module is to establish a secure connection between two devices and while establishing a connection authenticate user. Hence the security of the session would be maintained.

Screen Sharing Module-Once the user is authenticated a connection is established this module takes access of the server screen. Meanwhile this module takes control of the mouse and keyboard activity of the client side using an API. Once the access is taken, the communication between server and client starts, where the image feed is continuously exchanged with client and activity data are sent to server to reflect the activity. The image feed of the server is showed using windowing system on the client side which can be used in full screen mode as well. The communication between server and client is full duplex, and is implemented using threads to enhance performance of the application. One thread is responsible for sending image data to client and other responsible for receiving the activity data from client and replicate similar type activity on server side.

File Transfer Module-This module is responsible for allowing users to initiate and transfer files between the connected devices. To do this FTP connection is established between two devices. User just has to drag and drop the file to the window to transfer a file. If the connection is not established then user will be asked to establish the connection before transferring the file.

Video Transmission-This module is used to initiate transfer of the video files that can be watchable on the server side. It follows FTP protocol to initiate and complete the file transfer between two devices. It is very necessary that the connection is established between the devices before the file transfer is initiated.

Remote Control -This class is mainly used to listen to the activity of the client side. This module is responsible for capturing mouse and keyboard activity on client side and transferring it over to the server side. And this class is also responsible for replicating that activity on server side as well.

Multiple Clients-Handles various sessions established by many devices and makes session secured and compatible. Likewise this, all the six moduled mentioned above are used to control various operations on the application interface that allows used to share and control the desktops. This architecture is efficient and optimized solution to the remote desktop sharing.
A Security mechanism will be added where the server will be restricted to access all the sessions that the client is functioning.That is the server can view and control only those sessions which will be granted by the client.

IV. Experimental Results

Initial handshaking which establishes the connection between client and server. After connection, the server can access and control the screen of the client. Here, the Remote Buffer Protocol is used where image is send pixel by pixel. While updating the image at the client side we have to send only those pixels whose RGB values have changed. This would result in efficient use of bandwidth. Till now we are sending whole images at once. Then FTP protocol is also used by our application to perform file transfer between server and client. We have implemented drag and drop feature to make file transfer even simpler. Video streaming is implemented and after the whole project is implemented, server is connected with multiple clients using multiple sockets.

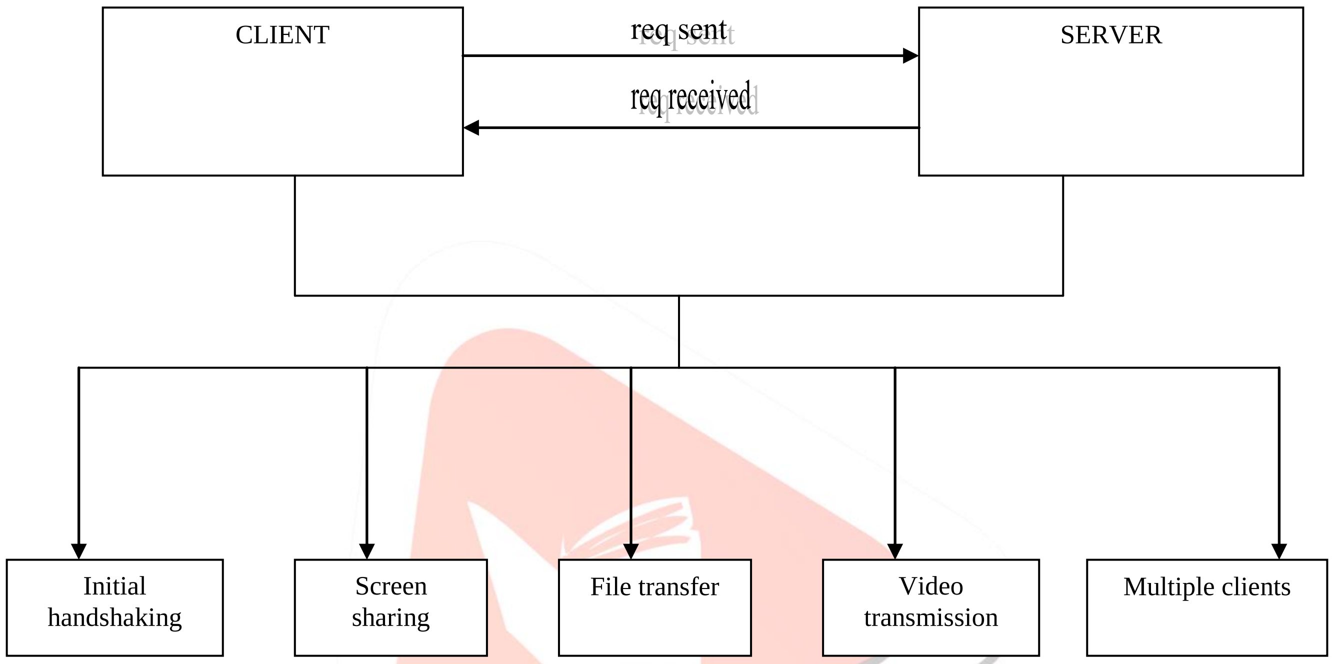

Figure 1 System Architecture

V. Conclusion And Future Work

We find that in our application a client can request the server for one or more specific services such as screen sharing, remote control, video streaming and file transfer. The performance of each service depends on the network characteristics such as bandwidth and distance from client to host. Screen sharing uses a modified version of basic RAW encoding so it also depends on the resolution of client and server.

It works best when resolution at both ends is same. File transfer service works fine for any file format. Video file only support java compatible formats. Video service also depends on the buffer size selected by us, the more the buffer size the more time it takes to stream but produces no sound. Since our application is totally java based so it is platform independent. We tested it on windows OS but it works fine for Linux also. The most striking feature of our application is that client can use software without having to install them. During run time of the application the server itself can also control its desktop. This helps the person sitting at the server side to show the client whatever he/she wants to. We conclude that application as for now works fine but can be improved through little modifications. The main modification of our code would be to add an audio streaming functionality.

The client requests the audio file to be played to the server. Then the server provides appropriate sessions for transmitting the audio while on the other hand the client receives this audio data through sessions. Another work to do is to provide valid authentication by the server of the client. The client can first register itself at the server and receive its username and password. The next time client tries to initiate the connection it will be asked by the server for its username and password which is send to the server for verification.

Providing an appropriate GUI to the application in which the client can:

- play any song by double clicking it.

- ask the client for services that it wants to receive.

REFERENCES

[1] JAVA NETWORKING PROGRAMMING By Elliotte Rusty Harold Published by O'Reilly Associates..

[2] Ichimura, S. Matsushita, Y. Sch. of Comput. Sci., Tokyo Univ. of Technol, "Lightweight Desktop-Sharing System for Web Browsers", IEEE 3rd International Conference on Information Technology and Applications, pp. 4-7, July 2005.

[3] Tae-Ho Lee Hong-Chang Lee Jung-Hyun Kim Myung-Joon Lee Sch. of Comput. Eng. & Inf. Technol., Ulsan Univ., Ulsan,"Extending VNC foe Effective Collaboration", IFOST 3rd international forum on Strategiv Technologies, pd 23-29 june 2008.

[4] http://www.microsoft.com/windowsmedia

[5] www.Ieeexplore.ieee.org

[6] http://www.uk.research.att.com/vnc/

[7] www.wikipedia.org

[8] http://forums.sun.com/thread.jspa

[9] http://communication.howstuffworks.com/how-desktop-sharing-works1.htm

[10] http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/guide/RTPRealTime.html

[11] http://java.sun.com/products/java-media/sound/

[12] http://java.sun.com/j2se/1.3/docs/api/javax/sound/sampled/AudioFormat.html