

Approach to access the Remote Device in same network using Virtual Network Computing: An Accessibility Endeavor

Shiv Kumar Goel^[1]
Deputy HOD (Asst. Prof.), Dept. of
MCA
VES Institute of Technology
Mumbai, India
shivkumar.goel@ves.ac.in

Nikita Haryani^[2]
Student of MCA
VES Institute of Technology
Mumbai, India
nikita.haryani@ves.ac.in

Abhishek Jain^[3]
Student of MCA
VES Institute of Technology
Mumbai, India
abhishek.jain@ves.ac.in

Prasad Kuvalekar^[4]
Student of MCA
VES Institute of Technology
Mumbai, India
prasad.kuvalekar@ves.ac.in

Pramod Tiwari^[5]
Student of MCA
VES Institute of Technology
Mumbai, India
pramod.tiwari@ves.ac.in

Abstract — Cellular phones have shown an impressive progress in their functionality to a point where it is now possible to have cellular phones execute Java programs. As a result, cellular users all over the world are now able to read as well as write e-mail, browse Web pages, and play Java games by means of their cellular phones. This development has prompted us to put forward the use of a cellular phone as a device for remotely controlling computers. Here, we will draft the process to access the home screen of remote computer using Virtual Network Computing based architecture with the use of an android phone. The user will be able to access the screen of remote computer through a browser on the user's phone. The primary criteria for the connection to be established is that both the desktop and mobile phone should be in the same wifi network. The image of the screen is compacted before it is sent to the phone. There is a provision for the left and right mouse click on the client side browser. The prototype is already implemented using java in Net Beans Integrated Development Environment using java server pages (JSP) and tested on an android based smart phone and tablet.

Keywords— *Android Smart Phone, Tablet, Java Development Kit, Wi-Fi, Desktop, Windows.*

I. INTRODUCTION

The innovations in smart phones have brought an enormous modernize in the methodological field associated with cellular phones. Currently, smart phones are used worldwide and provide features which were previously provided by computer system [1]. In this paper, we describe the system which can provide access to remote computer system using a smart phone within the Wi-Fi network. This system is being implemented for Android based smart phones and tablet. The security is maintained by providing authentication using a password protected login screen at the client side. VNC architecture and VNC protocols are used for client and server dealings [2]. In the scope of remote control there are numerous projects and initiatives designed to allow remote control between devices. Remote control architectures powered by manufacturers cover only a part of the features required for an effective use, and usually are designed as internal solutions. The most widespread system planned to perform remote control of devices is Virtual Networking Computing. This system was designed to improve application testing systems in mobile devices owing to the lack of capitals in mobile devices and the great cost of test settings. The progress of android platform is evolving endlessly and it is growing on a larger extent.

Virtual Network Computing is a graphical desktop sharing system providing remote control via network onto an android based smart phone.

II. PROPOSED ARCHITECTURE



Figure 1- Conceptual Representation

It can control the computer system by using a graphical screen update from a server and capturing a mouse click on the client side and mirroring the same on the desktop. VNC

system is executed by means of RFB (Remote Frame Buffer) protocol to convey all information between connected devices. Transmission is running on one port with the help of TCP/IP protocol [3]. VNC system mainly requires two kinds of applications - server application for a machine under control and client which is an android based smart phone. Client side is known as viewer owing to the reason of its functionality. Server machine can be called as controlling machine. It is accountable for a variety of tasks such as viewing a shared desktop (or screen in general) and capturing and converting all user activity into the RFB protocol messages. Server, on the other side, must interpret every single one event received from client and inoculate it in itself. Server responses to graphic screen update request by sending back a desktop view to connected client. The cellular user is able to see and operate the desktop on the cellular phone.

It allows user to login to the server and start a session between Android Smartphone & server. It first captures the screen of the desktop and compresses the captured screenshot. This compressed image is sent to the Android device on the client side. At the client side, it captures the various gestures that are performed by user on Android Smartphone. Gestures that are captured can be swipe left, right, up, down or even single tap or double tap. It next converts the captured gestures into mouse movements & clicks on the remote desktop. In order to reflect all the gestures performed by user on the smart phone screen, it constantly updates the Android Smartphone Screen whenever any gesture is performed by the user.

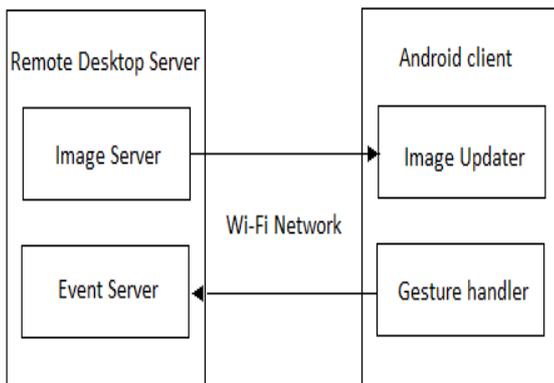


Figure 2- Architectural Representation

The above figure shows various components that are involved in Virtual Network Computing i.e. Remote Desktop Server and Android Client. Remote desktop server consists of Imager server and Event server. Android client consists of Image Updater and Gesture Handler. Detailed description about the components that are used in the system architecture is as follows.

Remote desktop server

It is windows based desktop or laptop running the server application. This server application is responsible for handling most of the processing. It has different components:

1. Image server: It is responsible for capturing the desktop screenshot image and store on the server machine.
2. Event Server: It is responsible for performing actual mouse events on the server machine according to the gestures performed over the Android smart phone.

Android client

The most of the processing takes place on the server machine so the android client only have to display the image received from the server and capture the touch gestures from the user. It has following components:

1. Image Updater: It just displays the image received from server on the Android screen. The image parameters must be altered depending on the Android device screen resolution.
2. Gesture handler: It captures the coordinates of the Android screen where the gestures were formed. This coordinates are send to the event server which performs the actual mouse events on the remote desktop.

The main functionalities that can be performed in the proposed system are described as follows:

A. Sharing of Remote Desktop Screen:

The remote desktop screen can be shared. VNC protocol makes it possible by using the concept of remote frame buffer (RFB). The protocol permits a server to update the frame buffer which is displayed on a viewer. It is applicable to all operating systems and applications as it works at the frame buffer level [4]. Any trustworthy transport such as TCP/IP can be used for functioning of the protocol. This module is responsible for authentication and connection that takes place between the client and server. It is achieved using the proper IP address of the server and the password once the handshaking is done. The VNC protocol (RFB) is very simple, based on one graphic primitive from server to client ("Put a rectangle of pixel data at the specified X,Y position") and event messages from client to server[5]. Writing a VNC viewer should not a tedious job, as it should be for any thin-client system. It involves only a reliable transport (usually TCP/IP), and a means of displaying pixels (either directly writing to the frame buffer, or going through a windowing system). Writing a VNC server is slightly harder than writing a client. The protocol is designed to make the client as simple as feasible, so it is generally up to the server to perform any necessary translations. For instance, the server must provide pixel data in the format the client wants. VNC is a client/server application. The server (which runs on the "target" machine that is, the machine whose desktop you wish to control) is called Vnc server and the client (cell phone) is called Vnc viewer. The VNC server and client communicate by means of a protocol called RFB (Remote Frame Buffer). The fundamental idea of this protocol is to communicate changes to the screen contents from the server to the client using a range of RFB encodings. In addition, RFB allows mouse and keyboard input on the client to be transmitted to the server, so that the client can not only passively observe the server's desktop, but dynamically interact with it.

B. Zooming Facilities:

Vnc client known as viewer can be zoomed in or zoomed out as and when required. The user can widen the viewport in order to browse its contents and also zoom in to view it in greater aspect[5].

C. Performing Mouse Pointer Events:

It facilitates the user to move the pointer on the viewer by pressing keys. The pointer can be moved on the remote desktop screen in both the directions vertically as well as horizontally. In order to state the start of the dragging operation, the user needs to press a key after that moving the pointer, and lastly pressing the same key to show the ending of the dragging operation. The viewport is automatically panned to follow the pointer, as the pointer reaches the border of the viewport. When we press the corresponding keys on the cellular phone, events like mouse button clicks are performed. Double-clicking is executed via pressing an exact key as a prefix.

D. Providing Input:

The user can enter the input with the help of incorporated text input facility of the cellular phone. Editing can also be performed using the same. The text needs to be transmitted to the VNC server, when the editing is done on the cellular phone. The similar technique can be used to send control characters like delete, backspace, carriage return, as well as line feed, by means of entering escape sequences. A new directory can be created on a remote UNIX system by entering the mkdir command with a carriage return as seven characters “mkdir\n” on a terminal emulator.

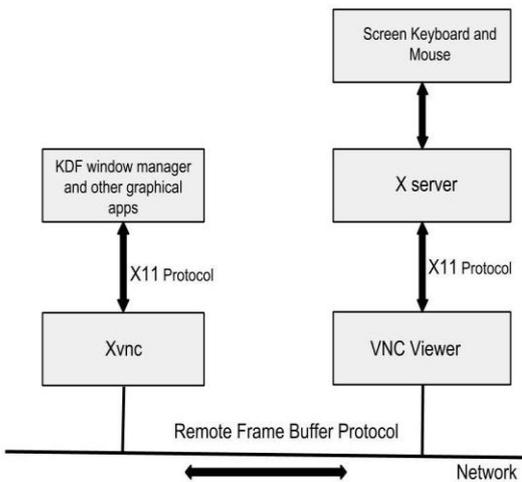


Figure 3 - VNC protocol using RFB

III. IMPLEMENTATION AND RELATED ALGORITHMS

Implementation is done in java using Socket Programming and Java Server Pages (JSP). Socket is responsible for providing the communication mechanism connecting two computers using TCP. A client program can create a socket at its end of communication and tries to connect that socket to a server. When the connection is completed, the server creates a socket object at its end [5]. The client and server

are now ready to communicate by writing to and reading from the socket. The java.net.Socket class corresponds to a socket, and the java.net.ServerSocket class provides a means for the server program to listen for clients and set up connections with them. The following figure shows the overview of communication between client and server using sockets.

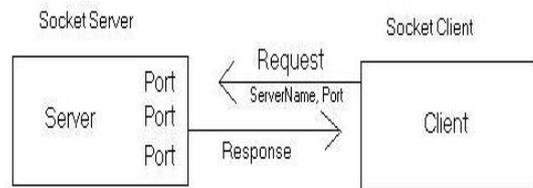


Figure 4 – Socket Programming

Java Server Pages (JSP) is a new technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types [6]. Java Server Pages is similar to PHP, but it uses Java programming language. JSP can be viewed as a high-level abstraction of Java servlets. JSPs are internally translated into servlets at runtime; each JSP's servlet is cached and re-used until the original JSP is customized. JSP can be used independently otherwise as the view component of a server-side model–view–controller design, usually with JavaBeans as the model, Java servlets as the controller.

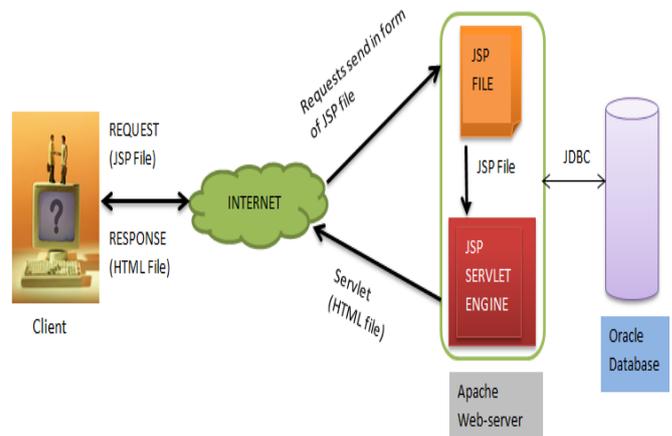


Figure 5 – Java Server Pages

Java Server Pages permits Java code as well as certain pre-defined actions to be interleaved with static web markup content, the resulting page being compiled and executed on the server to deliver a document. The compiled pages use Java byte code rather than a native software format. Similar to any other Java program, they must be executed within a Java virtual machine (JVM) that integrates with the server's host operating system to provide an abstract platform-neutral environment.

Algorithm for Multi Server Socket

1. Start
2. Import all necessary packages (i.e. java.net and java.io etc.)
3. Initialize ServerSocket object.
4. Specify valid port number.
5. Continue to listen on the port for connection from client.
6. If client connects then accept the request.
7. Throw an exception if unwanted behavior occurred.
8. End.

Algorithm for Client Socket

1. Start
2. Import all necessary packages (i.e. java.net and java.io etc.)
3. Initialize Socket object.
4. Specify host name and port number to whom client wants to connect.
5. Attempt for connection.
6. If connection is successful then start and start the image and event server threads.
7. Otherwise throw an exception.
8. End

Algorithm for Image Server

1. Start.
2. Import all necessary packages (i.e. java.net and java.io etc.)
3. Initialize ServerSocket object.
4. Wait till client to connect to the imgPort for transmission.
5. If client connect then start capturing the desktop screen.
6. Store the image on server machine then transfer it to the android Phone.
7. Update the image continuously.
8. Throw exception if unwanted behavior occurred.
8. End.

Algorithm for Event Server

1. Start.
2. Import all necessary packages (i.e. java.net and java.io etc.)
3. Initialize ServerSocket object.
4. Wait till client to connect to the evtPort for transmission.
5. If client connect then start capturing the gestures received from Android phone.
6. Perform the mouse events based on the gestures.
7. Throw exception if unwanted behavior occurred.
8. End.

Software Requirements

The software requires following things installed on the server machine:

- JDK 1.6 or above.
- NetBeans IDE 7.0 or above.
- Apache Tomcat Server 6 or above.
- MySQL 5.1 or above.

Hardware Requirements

- Wi-Fi Router.
- Android device (with version 2.1 or higher).
- Any Windows computer or laptop.

IV. RESULTS

The algorithms were implemented using socket programming with related packages and classes. The whole implementation was executed and tested on an android based smart phone and the following screenshots were taken.



VNC based access to remote computer from Android Smartphones

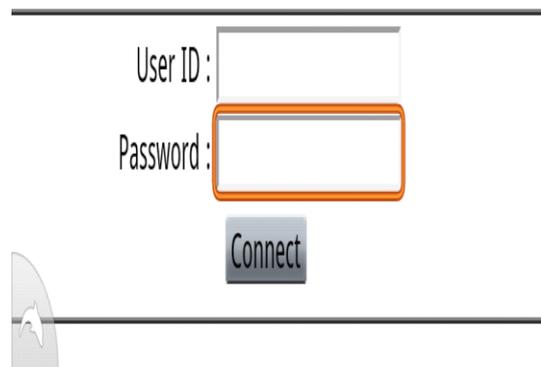


Figure 6- Login Screen



Figure 7- Display on Android Phone

REFERENCES

- [1] Md. Sanaullah Baig, Rajasekar M, Balaji P, "Virtual Network Computing Based Remote Desktop Access", Volume 3, Issue 5, May 2012
- [2] <http://www.ctxd.com/items/vnc-viewer.pdf>
- [3] <http://www.realvnc.com/docs/rfbproto.pdf>
- [4] <http://www.cl.cam.ac.uk/research/dtg/www/publications/public/files/tr.98.1.pdf>
- [5] [5]Angel Gonzalez Villan, and Josep Jorba, "Remote Control of Mobile Devices in Android Platform", arXiv:1310.5850v1 [cs.HC] 22 Oct 2013
- [6] http://en.wikipedia.org/wiki/JavaServer_Pages



Figure 8- Left click performed

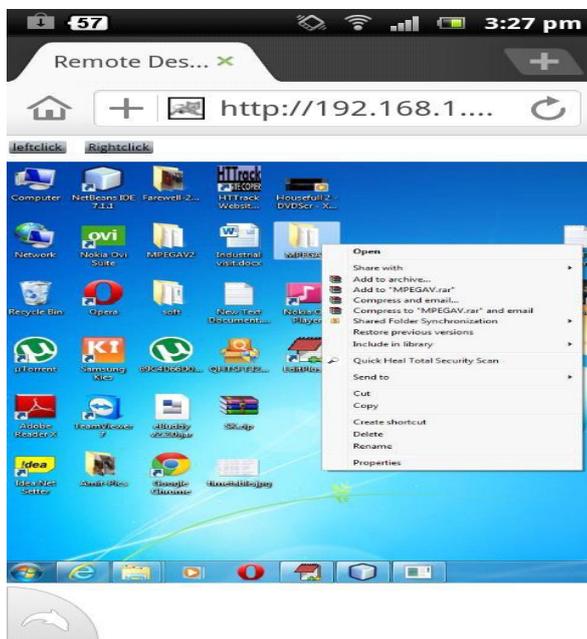


Figure 9- Right click performed

V. CONCLUSION

We have implemented a system to remotely access a computer desktop using Android Smartphone, despite the physical and bandwidth limitations of cellular phones. The system has architecture for accessing a remote desktop from the Android Smartphone. Android Smartphone user is able to view the remote desktop screen and control it using the touch gestures. The proposed system is portable thus allowing to control any windows based computer remotely. Currently, we are extending our implementation to support incremental updating of the sync viewer image, to speed up the frame rate and to incorporate more intelligent navigation. We are also trying to provide integrated panning and zooming of the viewport to simplify these basic operations, by applying speed-dependent automatic zooming.