ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

VNC ARCHITECTURE BASED REMOTE DESKTOP ACCESS THROUGH ANDROID MOBILE PHONES

Archana Jadhav[1] ,Vipul Oswal[2],Sagar Madane[3] ,Harshal Zope[4],Vishal Hatmode[5]

Department of Information Technology, Rajarshi Shahu College of Engineering,Pune-33,India.

ABSTRACT — ***In this paper, we will enlist the process to access the desktops of remote computer systems with the use of a android based cellular phone. This process will be carried out using Virtual Network Computing based architecture. A user will be able to access and manipulate the desktops of remote computers through a VNC viewer that will be provided on the user's cell-phone. Conditions that must be followed are that a VNC server must be installed on the person's computer which will be monitored and it must be connected to a Wi-Fi network. The user can access and manipulate the desktop within the Wi-Fi range irrespective of various platforms like windows, mac or linux. The image of the desktop is compressed before it is transmitted to the cellular phone. There are several functions provided so as to ease the viewing on cell-phones. There is shortcut function that can be used to quickly access the frequently used area. Current key assignments can be viewed using guidance function. A user can view two areas simultaneously using a twin view function. The prototype is already implemented using java and tested on a java based cellular phone.***

Keywords— **Android, Java, Wi-Fi, Mobile Terminal, Desktop.**

I. INTRODUCTION

The introduction of smart phones has brought a big change in the technical field related to cellular phones. Now a days , smart phones are used worldwide and provide much better facilities than previously available cellular phones. These phones provide features which were previously provided by computer system architecture. In this paper ,we describe the system which can provide access to remote computer system within the Wi-Fi network and provide features like desktop access, panning and zooming, over viewing ,twin view, file transfer. This system will be implemented on Android software stack. Android software stack[1] provides various packages for networking and it also provides high performance for android cellular mobiles. The security is maintained by providing authentication at the server side.VNC architecture and VNC protocols are used for client and server transactions. In the scope of remote control there are several projects and initiatives designed to allow remote control between devices. Although most of the architectures have the objective of control remotely PCs, there are some initiatives that aim to control mobile 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.

Other aspect to be considered is the remote visualization mechanisms that are useful for achieve a remote display other devices. The most popular system designed to perform remote control of devices is Virtual Networking Computing [3]. There are a large number of implementations to this solution including applied to Android software stack.This system was designed to improve application testing systems in mobile devices due to the lack of resources in mobile devices and the high cost of test environments. Also the solution proposed could be used to perform remote configuration. As part of the Android platform exists the Android Debug Bridge (ADB) protocol [1] to provide debug functionality on devices.This protocol is integrated in the platform and offers a service of server when is configured on the device. To manage the communication with this protocol, the Android Development Kit covers the ADB Client tool. Some of the features of this tool are the installation and de-installation of applications, downloading and uploading files, opening a shell console, starting applications, etc. The development of android platform is evolving continuously and it is expanding on large scale. This paper focuses on the control of Android platforms. This platform is an open source. In this paper Secction I describes the introduction about the android and brief idea summary of the system.Section II describes the architecture.

II.PROPOSED ARCHITECTURE

Cellular phones have shown a dramatic improvement in their functionality to a point where it is now possible to have cellular phones execute Java programs. As a result, cellular users throughout world are now able to read and write e-mail, browse Web pages, and play Java games using their cellular phones. This trend has prompted us to propose the use of a cellular phone as a device for remotely controlling computers. Virtual Network Computing is a graphical desktop sharing system providing remote control via network. It supports a controlling functionality by usage of a graphical screen update from a controlled device and capturing a mouse and/or a keyboard. VNC system is based on RFB (Remote Frame Buffer) protocol [3] to transmit all information between connected devices. Transmission is running on one port from range 5900-5906 using TCP/IP protocol.VNC system required two type of application for a proper work - server application for a machine under control and client - for a supervisor (controlling) device. Client side is called viewer because of its functionality. Controlling machine is responsible for viewing a shared desktop (or screen in general) and capturing and converting all user activity into the RFB protocol[7] messages. On the other side, server must to interpret all events received from client and inject them into self system. Server should also response to graphic screen update request by sending back a desktop view to connected client. The cellular user can see and manipulate the desktop on the cellular phone. The same cellular phone to talk someone, the user must terminate the network connection.

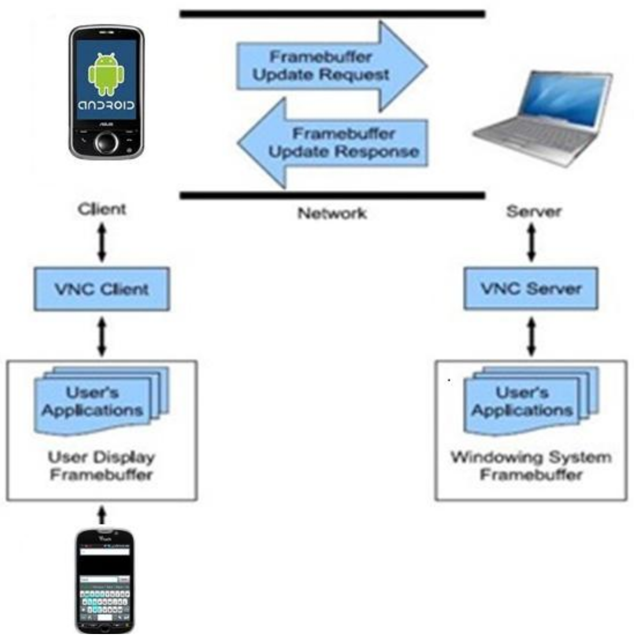

Fig.1. VNC architecture based mobile access computers

III. DESIGN OF THE PROPOSED SYSTEM

Here we have five main modules. Those modules are listed as below and the functionality and design of each module is described further.

A. Desktop Sharing:

In this module the remote desktop screen will be shared.this can be implemented with the help of the VNC protocol. VNC protocol is based on the concept of a remote frame buffer(RFB). The protocol simply allows a server to update the frame buffer displayed on a viewer. Because it works at the frame buffer level it is potentially applicable to all operating systems, windowing systems and applications. The protocol will operate over any reliable transport such as TCP/IP. This module delas with the authentication and connection between the client and server.The IP address of the server and the password transmits the first frame after handshaking is done.Writing a VNC viewer is a simple task, as it should be for any thin-client system. It requires only a reliable transport (usually TCP/IP), and a way 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 for a number of reasons. The protocol is designed to make the client as simple as possible, so it is usually up to the server to perform any necessary translations. For example, 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 want to control) is called Vnc server, and the client (cell phone) is called Vnc viewer. The VNC server and client communicate using a protocol called RFB (Remote Frame Buffer). The basic idea of this protocol is to communicate changes to the screen contents from the server to the client using various RFB encodings RFB also 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 actively interact with it.

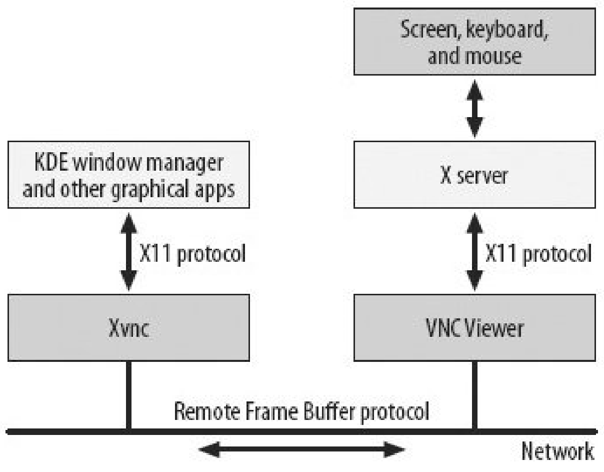
Screen, keyboard, and mouse
KDE window manager and other graphical apps
X server
X11 protocol
X11 protocol
Xvnc
VNC Viewer
Remote Frame Buffer protocol
Network

Fig 2: Basic Model of VNC protocol

The actual VNC protocol[4] is as described in the following section. The Xvnc server program acts as an X server. It accepts connections and X11 protocol messages from graphical applications just as any other X server would. The difference is that a "real" X server has a real frame buffer behind it (i.e., a real graphics display), whereas Xvnc has only an in-memory frame buffer. It accepts connections from VNC viewers and tells the viewer about any changes to its in-memory frame buffer. The viewer, in turn (mobile handset), connects to its own X server to actually get the image drawn on its screen. This architecture is shown in above figure.

B. Panning and zooming:

The user can move the viewport horizontally and vertically. The viewport can be widened (zoom out) to browse its contents and narrowed (zoom in) to see the display in greater detail.[1]

C. Over viewing and twin view:

In order to browse the entire area of the desktop display and to choose a specific area within it, the over viewing mode is provided. When the user turns this mode on, the aspect ratio is changed so that the whole area is rendered to fit the screen of the cellular phone.This helps the user adjust the viewport to the desired area of the desktop display.[1] Sometimes, it is convenient to display two areas of the desktop simultaneously. We can enter test conditions and observe the results simply by moving our line-of-sight slightly. Thus, we can quickly acquire a condition that produces the desired result. However, on a VNC client viewer, even though we can use shortcuts we still have to press keys many times to enter the test conditions and view the results repeatedly. To solve this problem, the VNC client viewer provides a Twin view. This view divides the display area of the cellular phone into two parts. The user can control the upper half and the bottom half independently. Therefore he can assign both halves independently to facilitate his desired task.[4]

D. Pointing and clicking:

The user can move the pointer on the remote desktop display vertically and horizontally by pressing keys. Dragging can be executed by pressing a key to specify the start of the dragging operation, then moving the pointer, and finally pressing the same key to indicate the end of the dragging operation. When the pointer approaches the edge of the viewport, the viewport is automatically panned to follow the pointer. Clicking mouse buttons can be performed by pressing the corresponding keys on the cellular phone. Double-clicking can be executed by pressing a specific key as a prefix.[4]

E. Inputting text:

Text is entered and edited locally on the cellular phone using the built-in text input capability of the cellular phone. After editing on the cellular phone, the text is transmitted to the VNC server. The same mechanism can be used to send control characters such as backspace, delete, carriage return, and line feed, by entering escape sequences. For example, the user can list the files on a remote UNIX system by entering the ls command with a carriage return as four characters “ls\n” on a terminal emulator.[4]

F. Shortcut Assignment:

Common GUI operations, such as pressing GUI buttons and opening pull-down menus, become very tiresome when only basic operations are provided. For example, to push a GUI button that is not currently displayed on the viewer, the user has to zoom out, pan several times, and may then have to zoom in to show the button. To shorten the time necessary to access frequently used display areas, the VNC client viewer offers a mechanism called a shortcut.

IV.ENCODING STUDY

The RFB protocol[3] working consisting of responding to request from the client about a specific onscreen rectangle and then server responds in the form of the update consisting of an encoding the difference between the moment of the request and the last time the client requested data about this rectangle .Sending of information will lead to the high consumption of the bandwidth with the consequent delay in the process. To overcome this problem different encodings have been developed. Encoding refers to the format in which a rectangle of pixel data will be sent. Every rectangle of pixel data is prefixed by a header providing the position of the rectangle on the screen, the width and height of the rectangle, and an encoding type. This encoding type specifies the way of encoding of the pixel data. The data itself then follows using the specified encoding. These encoding are used to determine the way to transfer the graphical information. Adding new encodings developed by third parties does not compatibility with VNC applications that do not contain that new encoding. When a client wants to establishes communication with server both side must negotiate the encoding type to be use. If the client requires a non-existent encoding, the server will appropriate the next encoding available.

A. Following are the 4 basic types of VNC encoding can be used:Raw, RRE, Hextile, Zlib and Tight.

1.) RAW:

RAW is the simplest encoding form. In this server sends all graphical pixels to the client and data consist in the form of width*height pixel values(where width and height are the width and height of the rectangle). This encoding method must be supported by clients. The process time used is minimal and the performance is very high when the server and the client are on the same machine. If the client is hosted in a remote device the performance is reduced due to the transfer of large amounts of data. This encoding is specifically designed for the low performing devices.[4]

2.) RRE:

RRE stands for Rise-and-Run-length-Encoding which consists of grouping consecutive identical pixels in order to send only the information of one pixel and the number of replications of that pixel. The basic intension behind RRE is the partitioning of rectangle of pixel data into subregions each of which consist of the single pixel value and union of which comprises the original rectangular region. It is an most effective method when large blocks of the same colour exist, like in patterns are to be send. There is a variant of the protocol which uses a maximum of 255x255 pixels to reduce the size of the packages. RRE rectangles arrives at the client in the form which are easily rendered immediately and efficiently by simplest of any graphics engines.[3]

3.) CopyRect encoding (copy rectangle):

This encoding is a very simple and efficient encoding. Which is used when the client already has the same pixel data elsewhere in its frame buffer. The encoding on the wire simply consists of an X,Y coordinate. This gives a position in the frame buffer from which the client can copy the rectangle of pixel data. This can be used in a variety of situations, the most obvious of which are when the user moves a window across the screen, and when the contents of a window are scrolled. A less obvious use is for optimizing drawing of text or other repeating patterns.[3]

4.) Hextile:

This encoding divides the rectangles in the 16*16 tiles, allowing the dimensions of the subresctables to be specified in 4 bits each and 16 bits in total. The rectangle is split into tiles starting at the top left going in left-to-right, top-to-bottom order. This encoded data contents of the tiles simply follow one another in the predetermined order. If the width of the whole rectangle is not an exact multiple of 16 then the width of the last tile in each row will be smaller than that of previous tiles. Each tile is either encoded as raw pixel data, or as a variation on RRE. Each tile has a background pixel value, as before. No need to specify for any given tile if it is same as background of the previous tile.[3]

V. APPLICATIONS

As VNC supports remote system administration it has lot of applications that includes system administration, IT support and helpdesks. It can be used not only for small scale purposes but also for various enterprise applications. As it supports multiple connections it can be used effectively for collaborative work. For example, it can be used for educational purposes for example students in a distributed group can view the computer screen which is been manipulated by the instructor. In this context the VNC architecture will be used by the android application for remote administration at the time of practical exams by the supervisor. Hence it is been used for many diverse applications.

VI. RFB PROTOCOL IMPLEMENTATION STEPS

There are three stages to the RFB protocol implentation. First is the handshaking phase, the purpose of which is to agree upon the protocol version and the type of security to be used. The second stage is an initialisation phase where the client and server exchange *ClientInit* and *ServerInit* messages.The final stage is the normal protocol interaction. Handshaking begins by the server sending the client a *ProtocolVersion* message. This lets the client know which is the highest RFB protocol version number supported by the server. The client then replies with a similar message giving the version number of the protocol which should actually be used (which may be different to that quoted by The *ProtocolVersion* message consists of 12 bytes interpreted as a string of ASCII characters in the format "RFB xxx.yyy\n" where xxx and yyy are the major and minor version numbers, padded with zeros.the server). A client should never request a protocol version higher than that offered by the server.

A. ClientInit:

Shared-flag is non-zero (true) if the server should try to share the desktop by leaving other clients connected, zero (false) if it should give exclusive access to this client by disconnecting all other clients.

B. ServerInit:

After receiving the *ClientInit* message, the server sends a *ServerInit* message. This tells the client the width and height of the server's framebuffer, its pixel format and the name associated with the desktop.

C. Client to server messages:

The client to server message types defined are:

Number	*Name*
0	*SetPixelFormat*
1	*SetEncodings*
2	*FramebufferUpdateRequest*
3	*KeyEvent*
4	*PointerEvent*
5	*ClientCutText*

Table 1:Client Server Message Types

VII.EXPERIMENTAL RESULTS

Experimental results describe result of the Remote Desktop Access system, in the form of input-output. i.e., it shows, after giving the specific input to the system, what kind of output is provided by the system, to the user.

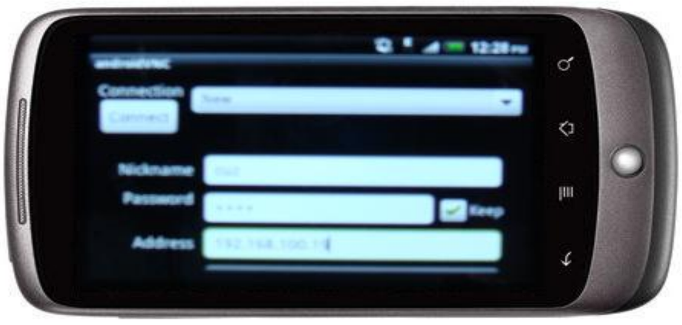

Fig.3 VNC Client GUI

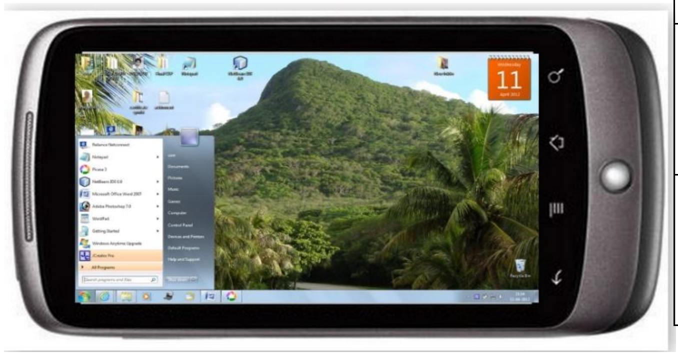

Fig.4 Desktop Access

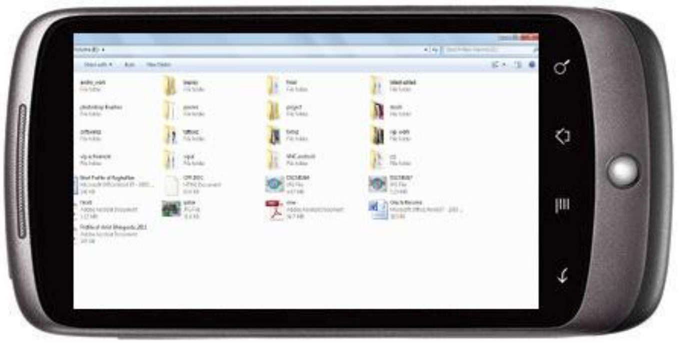

Fig.5 Panning Desktop

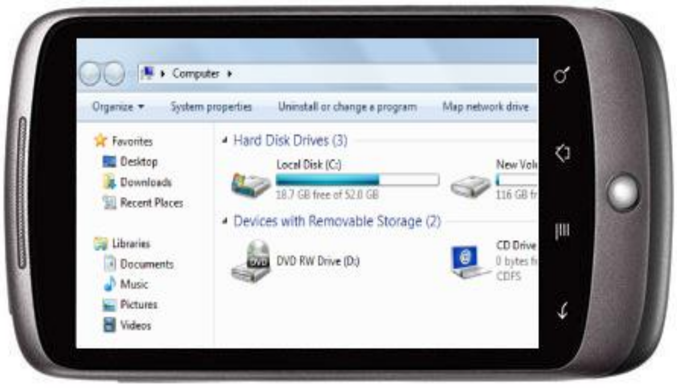

Fig.6 Zooming

INPUT	OUTPUT
User inputs username /nickname,password,ip address and port	User is logged into the system.
User can move the cursor	Accordingly the cursor position is changed on the desktop.
User can click	Accordingly the program is executed.
User can zoom	Depending on the area of zooming, the enlarged view of a region can be seen.
User can use shortcut assignments	Depending on the input related to a shortcut corresponding output is executed.

Table 2: Inputs and their Corresponding Outputs

VIII. Conclusion

This application will provide assistance to the system administrator in monitoring the tasks and also provide file transfer. Currently the scope of this system is within Wi-Fi area. Next step will be implementing this system over Internet. The same RFB protocol will be used for the data transfer. The VNC architecture will be used for implementation of the system. Due to wide use of android devices, this system will be developed for tablets and other handheld devices. This system will provide mobility for users for controlling their computer desktops over internet. More facilities and features for accessing applications running on remote desktop from mobile handheld devices will be provided. Thus the extended scope of this system will prove to be helpful in providing mobility and accessing the remote desktop over the internet.

REFERENCES

[1] Android. http://www.android.com Retrieved March 1st, 2011.

[2] Remote Control of Mobile Devices in Android Platform Angel, Gonzalez Villan , Student Member, IEEE and Josep Jorba Esteve,Member, IEEE.

[3] www.realvnc.com/docs/rfbproto.pdf, reviewed on June 20th,2011

[4] Virtual Network Computing,Tristan Richardson, Quentin Stafford-Fraser,Kenneth R. Wood and Andy Hopper,Reprint from IEEE Internet Computing Volume 2, Number 1 January/February 1998.

[5] Global Telecommunications Conference GLOBECOM 2010), 2010 IEEE.
[6] T. Richardson, \The RFB Protocol", Tech. rep., RealVNC Ltd,2007
[7] The RFB Protocol,Tristan Richardson,RealVNC Ltd(formerly of Olivetti Research Ltd / AT&T Labs Cambridge) ,Version 3.8,Last updated 26 November 2010
[8] A. P. Rajshekhar, Socket Programming in Java.
[9] Zhang Juan,"The design of intertranslation dictionary software of online access and desktop access",Electric Information and Control Engineering (ICEICE), 2011,International Conference, 15-17 April 2011.
[10] Dixit, S.,"Data rides high on high-speed remote access",Communications Magazine, Volume: 37, Issue: 1,Jan 1999.
[11] Muhammad Wannous, Student Member, Hiroshi Nakano, "NVLab, a Networking Virtual Web-Based Laboratory that Implements Virtualization and Virtual Network " Computing Technologies IEEE.

Archana Jadhav
Asst. Professor at
Department of Information Technology, Rajarshi Shahu College of Engineering, Pune-33,India.

Harshal S. Zope
Department of Information Technology, Rajarshi Shahu College of Engineering, Pune-33,India.

Vishal V. Hatmode
Department of Information Technology, Rajarshi Shahu College of Engineering, Pune-33,India.

Vipul V. Oswal
Department of Information Technology, Rajarshi Shahu College of Engineering, Pune-33,India.

Sagar M. Madane
Department of Information Technology, Rajarshi Shahu College of Engineering, Pune-33,India.