

Virtual Network Computing Viewer Using Remote Frame Buffer Protocol

Nandhini.S, Archana.N, Bagavathi.S, Arunachalam.M

^{1,2,3,4} Department of Computer Science and Engineering,

K.L.N College of Information Technology, Sivagangai, Tamil Nadu, India – 630611

Abstract: *Virtual Network Computing (VNC) is a remote control system which allows the user to view and interact with one computer (the “server”) to another computer or cellular phones (the “viewer” or client) anywhere on the internet. A viewer is provided on the cellular phone that enables the user to see and manipulate the desktop of various remote systems such as MS Windows, Macintosh, and UNIX. The system to be accessed must be running a VNC server and it must be attached to a network. A proxy is used to send the image of the desktop to the cellular phone, to convert different devices, to suppress network traffics, and to support recovery from an unscheduled disconnection. A prototype of the proposed system has to be implemented using Java and will be tested on android phone. Virtual Network Computing is a simple protocol for remote access to graphical user interfaces. It is based on the concept of a Remote Frame Buffer (RFB). Several functions to the remote computer or the server desktop such as viewing the desktop, mouse click operations, sending a text message, opening of documents or files, opening of the task manager, manipulating of files and several other functions can be performed from our cellular phone.*

Keywords: Virtual Network Computing, Remote Frame Buffer, Remote Control, Remote Desktop.

I. INTRODUCTION

Virtual Network Computing (VNC) is an ultra-thin client system based on a simple display protocol that is platform independent. VNC was originally developed at the Olivetti Research Laboratory in Cambridge, England. It achieves mobile computing without requiring the user to carry any hardware. The VNC software should able to communicate with the remote system and can find out whole description about the remote system and can have control on the peripheral devices from the cellular phones. The remote system administration tasks can also be performed from the cellular phone [1] and [2].

Ha-Young Ko et al [3] implemented a prototype system for mobile VNC, and several works were done for improving the screen update rate. They reported that JPEG is the most suitable for mobile VNC in terms of both complexity and compression ratio. In addition, the proposed modified region coding decreases the encoding time, and consequently increase screen update rate.

Daniel Thommes et al [4] proposed a new solution called RemoteUI, which worked with abstract user interface (UI) descriptions and their prioritized, incremental remote replication. The system combined different approaches from thin client computing and widget based UI systems and was further optimized in the context of wireless networks and mobile consumer

devices. By applying network-based user interface systems to control these devices, innovative applications became feasible.

In VNC system, server machines supplied not only applications and data but also an entire desktop environment that can be accessed from any internet-connected machine using simple software VNC. VNC provide mobile computing without requiring the user to carry any device whatsoever. In addition, VNC allowed a single desktop to be accessed from several places simultaneously, thus supporting application sharing in the style of Computer-Supported Cooperative Work (CSCW) [5].

In computing, VNC is a graphical desktop sharing system which uses the protocol to control the computer remotely, but now according to our methodology remote computers can be accessed from cellular phones. It transmits the keyboard and a mouse event from remote computer to cellular phones. The objective of our work is to develop an application through which user can monitor the computer over a network and can perform the administrative tasks from the cellular phones itself. The rest of the paper has been organized as follows: Section II proposes the novel approach and experimental results and Section III concludes the paper.

II. PROPOSED METHODOLOGY

Authentication is set by asking the username and password on the android device. If the validation is correct, it asks for the IP address of the remote system. Otherwise, it displays an error message. If the IP address is incorrect it displays an error message and if the IP address is correct. With the help of the HTTP proxy, the client is connected to the server which is our remote desktop. Connection from the server to the mobile client can be done using Stream Socket Connection. As the connection is wireless we should use internet with GPRS activated android device which is compatible for the java programs to run.

HTTP supports proxy server. A proxy server is a computer that keeps copies of responses to recent requests. In the presence of a proxy server, the HTTP client sends a request to the proxy server. The proxy server checks its cache. If the response is not in the cache, the proxy server sends the request to the corresponding server. Incoming responses are sent to the proxy server and stored for further requests from client. The proxy server reduces the load on the original server, decreases traffic and improves the latency. Proxy is used to send the

image of the desktop to the android device. Proxy is also used for the compatibility of the different devices to run the java programs.

The computer screen's view is fragmented into 8 to 16 parts to synchronize with the display area of the mobile using Remote frame buffer protocol(RFB).A VNC server sends a remote desktop display as bitmap images in RFB protocol. A SVNC proxy converts the display image and then transfers the converted image to a SVNC viewer in response to a user request that was received from that SVNC viewer. The transfer is performed in our own Compact RFB (CRFB), our simplified RFB protocol. Then, the SVNC viewer displays the transferred images Menu with various methods which will be displayed on the android device.

When android-VNC-viewer is opened, Connection configuration page will be viewed. Set up a connection with a VNC Server, or choose an already configured Connection. The page is divided into two Sections. The top section contains a drop- Down list of the connection configurations. When android-VNC-viewer is run for the first time, this list will have only one entry: New. Select to create a new configuration (Fig 2.1) for connection configuration.

The Connect button will start the VNC client with the currently selected configuration. The bottom section of the page is page with connection configurations. There are more options that will fit on the page, so this section scrolls.

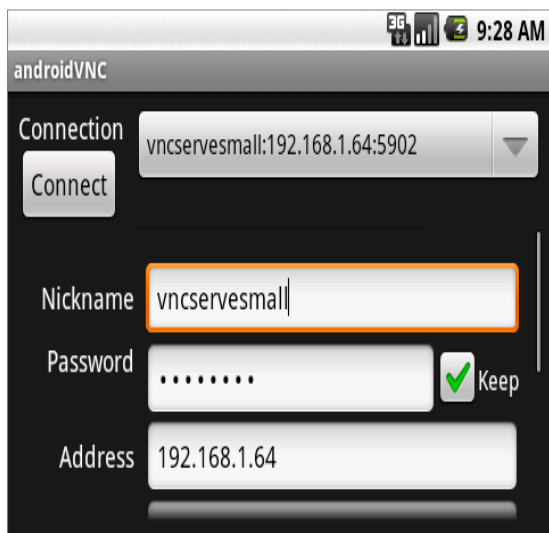


Fig 1 Connection configuration (a)

2.1 Nickname: Each configuration has a Nickname which will make it easy to find in the list.

2.2 Password: If the VNC server is configured to require a password, enter the password here.

2.3 Address: DNS name or IP address of the computer running the VNC server to which the user wants to connect, must be entered in this field.

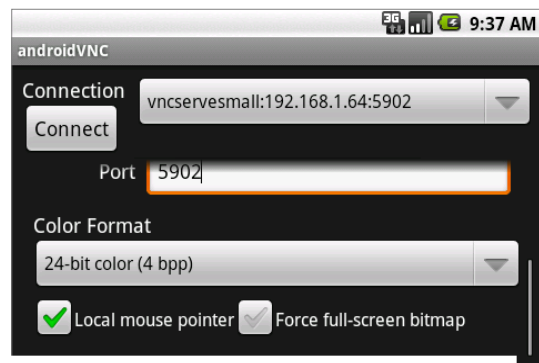


Fig 2: Connection configuration (b)

2.4 Port: Port number of the VNC server should be entered here. This field will default to the first VNC port number, 5900. **2.5 Color Format:** The client supports a number of color formats, which are specified by number of colors/number of bits per pixel.

2.6 Use local mouse pointer: Checking this option will cause **android-VNC-viewer** to draw a small square cursor at the mouse position (it's not the actual mouse cursor, but it's better than nothing). Refer fig 2.2 for viewing connection configuration dialog box.

2.7 Force full-screen bitmap to work with these large displays, **android-VNC-client** breaks large display into smaller tiles and works with only one tile at a time. Unfortunately, this will create some visual artifacts. In particular, when a large display is in Fit to Screen mode only a portion of it will be visible, and in 1:1 mode as the user can pan over a large display and can see different parts of the display flicker in.

2.8 Repeater: Press this button if the user is using an UltraVNC repeater to connect to VNC server. In this case, the user should have entered the address and the port of the repeater in those fields; in the repeater dialog the user should enter the VNC server address from the repeater and the screen id or port number, separated by colons.

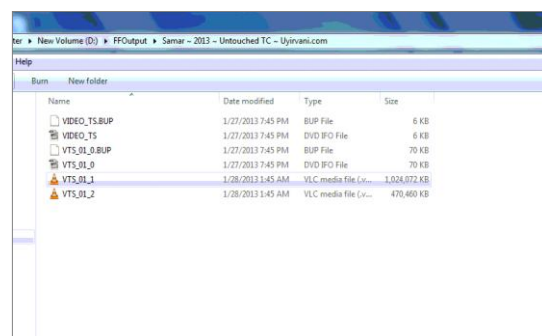


Fig 3: Viewing files

By using this application the user can view various types of files that are available in the system from user's android device itself. Various types of files that are accessible are text files, word files, excel files, audio files, video files, image files, etc. The figure 2.3 shows

the screenshot of an android device containing various types of files that are available in user's system.



Fig 4: Special controls

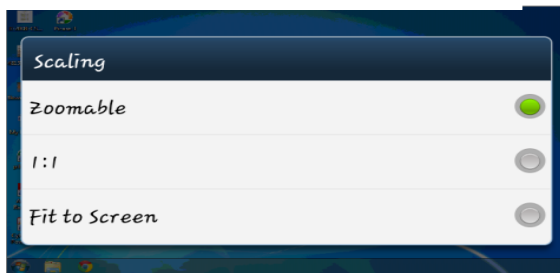


Fig 5: Scaling Menu

The served display can be shown on the Android device in three modes. The user can switch between the modes with the Scaling (Menu-Z) menu item. See fig 2.5 for viewing scaling options.



Fig 6: Zoom able

Zoom able is the default mode with the latest version of **android-VNC-viewer** (Fig 2.6). It allows selecting one among the zoom levels on the fly, so that the user can see the screen or zoom in for more accurate touch control.

1:1 mode shows the display with pixels mapped one-to-one. Typically, only a portion of the served display will show on the Android device; which allows panning around the display with the device controls. **Fit to Screen** mode scales the full display so it will fit on the device screen. This doesn't work properly with large displays, since the phone can't fit them all into its memory.

In Input Modes, there are several modes for adapting the input controls of the Android device to the VNC display, to enable a style of interaction most convenient for the applications the user is using.

Touch Mouse Pan and Zoom is the default input mode and is designed to work like the Android browser. The user can control both pan the display and control the mouse using the touch screen and gestures.

Touchpad Mode is the alternate gesture-based interface. In this mode, touching the screen moves the mouse cursor like a mouse touchpad. The screen will pan to follow the mouse (unless user turns it off). Tapping the screen clicks the mouse, where it is rather than where it is tapped. For multi-touch capable devices, drag two fingers to pan the screen independent of the mouse position. Otherwise, it works like Touch Mouse Pan and Zoom.

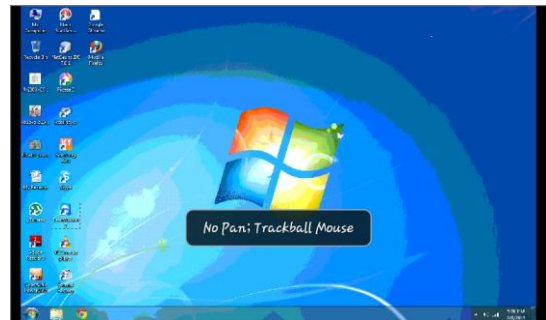


Fig 7: No pan, Trackball mouse mode

No Pan-Trackball Mouse mode is only available in Fit to Screen scaling and is the only input mode available then. In this mode the touch screen is not used. Keyboard events are sent to the server and the trackball (if the user's device, like the G1, has a trackball) controls the VNC mouse. The figure 2.7 shows the screenshot of no pan; trackball mode.

Desktop Panning Mode is the mode where both the touch screen and the trackball are used to pan the device display over the larger VNC display. Keyboard events are sent to the server. Pressing the trackball toggles between Desktop Panning and Mouse Pointer Control modes.

Mouse Control Mode uses the touch screen to control the mouse. Touching the screen generates a mouse click at that point; dragging on the screen creates a mouse drag.

Keyboard events are sent as normal. The trackball is used to send arrow-key events to the VNC server. Pressing the trackball toggle between Mouse Pointer Control and Desktop Panning modes.

Touch Pan; Trackball Mouse drags on the touch screen to pan the device display over the VNC display. Keyboard events are sent to the server. The trackball controls the VNC mouse. Pressing the trackball sends a mouse click; holding the ball down while rolling accomplishes a click and drag. This is the default input mode when scaling is set to One-to-One.

DPad Pan; Touch Mouse Mode uses the directional pad (available on some devices) to pan the display over the VNC display. Touch the screen to send a mouse click; touch and slide to send a mouse drag. Use the camera button while touching the screen to simulate a right-button click or drag.

Some **Special Controls** try to make up for missing features of an Android device (i.e., a G1) as a VNC controller. To use the right mouse button, hold down the **Camera** button while clicking as appropriate for the

selected input mode (touching the screen or clicking the trackball). If the phone has no dedicated camera button, right click can be done by double-tapping or with the `_Send Keys_` menu item.

To send the Esc key, press the **Back** button on the device. The Alt button generally is used to send the special symbols that aren't otherwise available on the keyboard, rather than sending the Alt meta-key over the VNC connection.

The volume control buttons simulate turning the mouse scroll wheel up or down. (Holding down doesn't work, to scroll further click repeatedly).

2.10 Other Menu Items

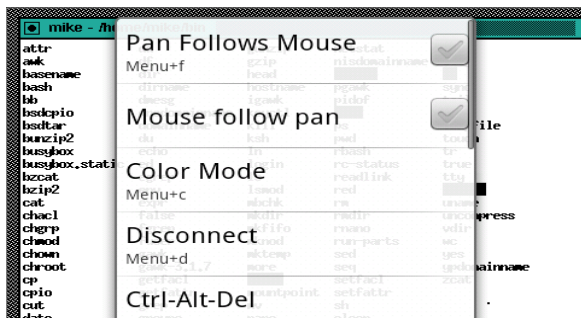


Fig 8: Menu items

- Send Keys (Menu-S) this opens the Special Keys dialog to lets the user to configure and send special keys and combinations. Mouse @ (Menu-M) Wraps the VNC mouse to center of the portion of the display shown on user's device. Send Text (Menu-E) opens a dialog that lets to enter a block of text that can be sent to the server (fig 2.8). Useful for phones without a physical keyboard. Color Mode (Menu-C) allows to switch the Color Mode of the connection.

In Pan Follows Mouse (Menu-F), the display will automatically pan to follow the mouse around the screen. In Mouse follow pan, the users pan the display. When the mouse moves off the visible portion of the screen, the mouse will be warped to the middle of the screen. The user can use both follow modes together. Disconnect (Menu-D) Closes the current VNC connection, returns to the configuration page.



Fig 9: Shortcut keys

- Ctrl-Alt-Del (Menu-A) Sends Ctrl-Alt-Del over the VNC connection (fig 2.9).

- Info (Menu-I) Gives information about current configuration

- Send Key Again (Menu-G) Send the last sent special key again

Multiple lists of special keys are supported. The user can use the buttons at the bottom of the scrolling section of the dialog to create a new list or to Copy user's list to a new name. This way the user can have compact, application specific lists for quick access to the keys user's need. An entry on the VNC Client menu , lets the user to re-send the last special key that has been sent with the shortcut Menu-G. User can operate this dialog from the keyboard as well as using the touch controls. Press Alt and Shift to toggle those modifiers. Press the Search button to toggle the Ctrl modifier. When the user type a regular key on the keyboard, key plus the selected modifiers will be sent immediately to the server. For device without keyboards, android-VNC-viewer can work effectively on devices without physical keyboards. Use the Send Keys menu option to send keystrokes that are commands in themselves (i.e. Alt-Tab). When the user needs to enter a block of text, use the Enter Text command, type the text with the Android keyboard, and press send. The user can edit the text before it is sent to make sure it is correct, and Enter Text command will remember the texts that user have sent in the session, so the user can send them again quickly (convenient for passwords that the user have to enter many times). For shortcuts android-VNC-viewer, lets the user to create home screen shortcuts for quick access to the favorite connections. Long click on the home screen, select Shortcuts in the menu, then scroll down to VNC Connection. The user will get a page list of configured connections. Select one and it will be added to the home screen as a short cut. Selecting a short cut starts the connection immediately.

III. CONCLUSION

VNC allows to view a computing 'desktop' environment not only on the machine where it is running, but from anywhere on the Internet and from a wide variety of machine architectures.VNC uses the protocols to remotely control another computer, whereas our methodology access to remote computers from android devices. It helps in controlling the peripheral device like performing the mouse click operations and manipulation of files. VNC allows to be connected in the network and will provide secure, easy access to information within the network. In our future work, we will work on the memory management and on the file sharing management. Memory management deals with managing and sharing memory among the systems that are connected to the android device. In file sharing, we are going to work on sharing the files stored in the android device to the system that we got connected.



ISSN: 2277-3754

ISO 9001:2008 Certified

International Journal of Engineering and Innovative Technology (IJET)

Volume 2, Issue 8, February 2013

REFERENCES

- [1] Md. Sanaullah Baig, Rajasekar M. and Balaji P “Virtual Network Computing Based Remote Desktop Access”, International Journal of Computer Science and Telecommunications, Vol. 3, No. 5, 2012.
- [2] Margaret Butler, “Android: Changing the Mobile Landscape”, IEEE Pervasive Computing, pp. 4–7, 2011.
- [3] Ha-Young Ko, Jae-Hyeok Lee, Jong-Ok Kim, “Implementation and Evaluation of Fast Mobile VNC Systems”, IEEE Transactions on Consumer Electronics, Vol. 58, No. 4, 2012.
- [4] Daniel Thommes, “RemoteUI: A High-Performance Remote User Interface System for Mobile Consumer Electronic Devices”, IEEE Transactions on Consumer Electronics, Vol. 58, No. 3, 2012.
- [5] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper, “Virtual Network Computing”, IEEE Internet Computing, Vol.2, 1998.