ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

Mobile Virtual Network Computing System

Vidhi S. Patel, Darshi R. Somaiya

Student, Dept. of I.T., K.J. Somaiya College of Engineering and Information Technology, Mumbai, India

ABSTRACT: we are planning remote access to Symbian OS based smart-phones. Well known remote protocol called VNC (Virtual Network Computing) is used to share phone screen and exchange events. In our project, we are looking forward to implement a prototype system for mobile VNC, and several works are done for improving screen update rate. In our project, we are looking forward to implement a prototype system for mobile VNC, and several works are done for improving screen update rate.

KEYWORDS: Virtual Network Computing, Remote Frame Buffer, Modified Region Encoding, Parallel Communication

I. Introduction

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 Japan 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. For example, if a cellular user is able to remotely access computers (such as workstations in offices and personal computers (PCs) in homes) or other networked digital appliances, it would provide the user with the following capabilities:

- to see the contents of a file placed on the desktop of a remote computer.
- to reboot a remote server as an administrator.

While it is not very difficult to develop a specific system to satisfy each of the above operations separately, it lacks the generality needed for performing several such functions with one device. This paper presents a virtual network computing(VNC) based architecture for accessing the desktop of various remote systems (such as MS Windows, Macintosh, and UNIX systems) from a cellular phone. It is assumed that the remote computer system is running a VNC server and that it is attached to a network. The cellular user can see and manipulate the desktop on the cellular phone.

II. Related work

Key issue of the topic is screen sharing, for sharing screen; we need to transfer image of screen from server to client's screen using limited network bandwidth. Previous works have studied appropriate screen encoding schemes for image to be sent over network. Along with speed, compression ratio and efficiency of encoding image according to receiver device also matters, Tight-VNC extended VNC by using data analyzer and a set of data filters as preprocessors to improve adaption ability in encoding. They also classified blocks of image into categories hence improved technique of compression. Categories likes variation in motion parts, low motion and high motion parts [1] encoded using run-length coding.

Some other works have focused on drawback of client-pull system. Screen is only updated when the client requests. This situation is feasible for low-latency environment, In high latency environment, request from the client can be delayed hence resulting in affecting screen update performance of the VNC.

III.SYSTEM DESCRIPTION

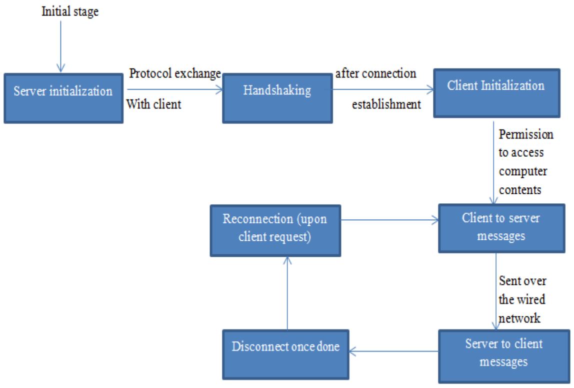
A) Mobile VNC System

In our project, We are planning remote access to Symbian OS based smart-phones. Well known remote protocol called VNC (Virtual Network Computing) is used to share phone screen and exchange events. Proposed VNC server will run on the Symbian smart-phone and any VNC client designed for PC can connect and share session with mobile devices. VNC (virtual network computing) is a popular tool for sharing an application, allowing users to access graphic displays remotely.

In mobile VNC systems, it has been challenging to increase screen update rate by fast screen image encoding. In our project, we are looking forward to implement a prototype system for mobile VNC, and several works are done for improving screen update rate. At first, a number of video encoders are integrated into a prototype system. The existing RFB protocol is extended straightforwardly to integrate video codecs.

Next, the overall system architecture is modified from serial operation to parallel. Finally, we propose a modified region coding to further reduce the encoding time of screen images. We report that JPEG is the most suitable for mobile VNC in terms of both complexity and compression ratio. In addition, the proposed modified region coding can decrease encoding time, and consequently increase screen update rate.

B) System Block diagram



C) System Algorithm

VNC session stages:

VNC session is divided into three main stages. The first one is “Handshaking”, then “Initialization” and after them normal process occurs (see fig. 2). In first phase remote machine establishes a connection and negotiate any special functionality of a protocol used during communication. Negotiations begin with a decision of a protocol version supported by both side. Then is an authentication stage and after that, other features are set during an initialization. During normal VNC session client sends only information of key or pointer events and asks to send him back a graphic buffer of a server desktop (screen). Server is responsible for handling all messages from client and generates a pixel buffer enclosed desktop area.

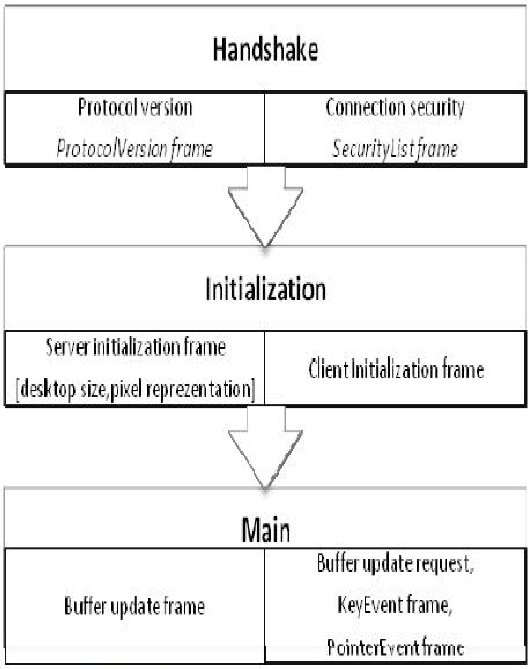


Figure 2. The three stages of VNC session.

To control a Smartphone using VNC system it is obligatedto implement a server side application. It is necessary to develop a server application providing functionality essential for the VNC communication and posses a possibility to interact with the Symbian OS environment. Control of all phone features is crucial to a server process under a remote command.

On the other hand server should work in background and avoid any kind of interaction with an environment if it is not necessary. Should not interrupt any working process and be transparent for them. Should minimize resources consumption and operating time used for a proper self-work. Working server should fulfill the requirements not to interact with other application when it is not needed. Furthermore, server should assure connection to network using WIFI technologies, which is available on many modern smart phones and be able to carry out requests received from connected clients.

D) COMPARISON BETWEEN EXISTING SYSTEM AND PROPOSED SYSTEM

EXISTING SYSTEM

There is an existing system that helps access the computer remotely with the help of mobile phone device. The operating system supported by the existing system is Symbian OS. It doesn’t allow for any other operating system to access the computer remotely. As it is difficult for the developer to develop the testcases in order to check every application that becomes expensive, in the existing system, computer system tests the mobile application with the help of some test codes and protocols that connects the computer and the mobile device remotely.

Several disadvantages in existing VNC that motivated us for enhancing are enlisted as follows:

- Serial Communication between Client and Server.

- Poor Encoding Schemes.
- Slower Screen Update Rate.

These reason lead to inefficient usage of network and bandwidth.

PROPOSED SYSTEM

Our Aim is to extend defined protocol. Three main extensions we are planning: new encodings, pseudo encodings and new security types. New encodings can be added to speed up communication (for example by using better compress algorithm) if encoding is not supported by server VNC messages are simply ignored. Pseudo encodings is used by client to inform server that is supports certain extension. If developer would like to add new protocol and even give up RFB new security types can be used.

- In our proposed system we can access that the remote computer using that the Java enabled Mobile phones. In this system we use cellular phones with internet connectivity and a personal computer. The desktop of the PC can be viewed on the mobile phone depending on the resolution of the mobile PC on the mobile. We can perform many operations such as viewing that Desktop of the PC, Accessing Applications in the PC, etc. The mobile is easily portable thereby portability exists also one can set security authentication. There by security is highly enhanced. In order to increase the speed of the pointed the speed can be enhanced, which exists. Also one can view the desktop either in normal and full mode. Navigation of screen is also possible.

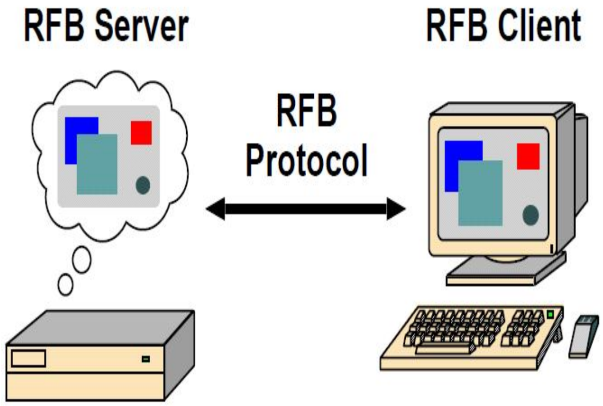
- To extend scope of existing System, We have a plan to utilize all the possible ports available from range of transmission. Hence extended use of ports will give us flexibility of parallel communication between server and client.

- With flexibility of parallel communication, we can easily boost the update rate of client screen. Further, we are planning to introduce concept of modified region encoding in our project.

- Our proposed project is limited to java platform, if time permits we are planning to implement on Android Platform.

VNC - 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 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 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.

RFB - ("*remote framebuffer*") is a simple protocol for remote access to graphical user interfaces. Because it works at the framebuffer level it is applicable to all windowing systems and applications. The remote endpoint where the user sits (i.e. the display plus keyboard and/or pointer) is called the RFB client. The endpoint where changes to the framebuffer originate (i.e. the windowing system and applications) is known as the RFB server. RFB is truly a "thin client" protocol. The emphasis in the design of the RFB protocol is to make very few requirements of the client. In this way, clients can run on the widest range of hardware, and the task of implementing a client is made as simple as possible.



The protocol also makes the client stateless. If a client disconnects from a given server and subsequently reconnects to that same server, the state of the user interface is preserved. Furthermore, a different client endpoint can be used to connect to the same RFB server. At the new endpoint, the user will see exactly the same graphical user interface as at the original endpoint. In effect, the interface to the user's applications becomes completely mobile. Wherever suitable network connectivity exists, the user can access their own personal applications, and the state of these applications is preserved between accesses from different locations. This provides the user with a familiar, uniform view of the computing infrastructure wherever they go. The display side of the protocol is based around a single graphics primitive: *"puta rectangle of pixel data at a given x,y position"*. At first glance this might seem an inefficient way of drawing many user interface components. However, allowing various different encodings for the pixel data gives us a large degree of flexibility inhow to trade off various parameters such as network bandwidth, client drawing speed and server processing speed. A sequence of these rectangles makes a *framebuffer update* (or simply *update*). An update represents a change from one valid framebuffer state to another, so in some ways is similar to a frame of video. The rectangles in an update are usually disjoint but this is not necessarily the case. The update protocol is demand-driven by the client. That is, an update is only sent from the server to the client in response to an explicit request from the client. This gives the protocol an adaptive quality. The slower the client and the network are, the lower the rate of updates becomes. With typical applications, changes to the same area of the framebuffer tend to happen soon after one another. With a slow client and/or network, transient states of the framebuffer can be ignored, resulting in less network traffic and less drawing for the client.

VNC session is divided into three main stages. The first one is "Handshaking", then "Initialization" and after them normal process occurs (see fig. 2). In first phase remote machine establishes a connection and negotiate any special functionality of a protocol used during communication. Negotiations begin with a decision of a protocol version supported by both side. Then is an authentication stage and after that, other features are set during an initialization. During normal VNC session client sends only information of key or pointer events and asks to send him back a graphic buffer of a server desktop (screen). Server is responsible for handling all messages from client and generates a pixel buffer enclosed desktop area.

Input Protocol

The input side of the protocol is based on a standard workstation model of a keyboard and multi-button pointing device. Input events are simply sent to the server by the client whenever the user presses a key or pointer button, or whenever the pointing device is moved. These input events can also be synthesised from other non-standard I/O devices. For example, a pen-based handwriting recognition engine might generate keyboard events.

VNC Sessions -

VNC session is divided into three main stages. The first one is "Handshaking", then "Initialization" and after them normal process occurs (see fig. 2). In first phase remote machine establishes a connection and negotiate any special functionality of a protocol used during communication. Negotiations begin with a decision of a protocol version supported by both side. Then is an authentication stage and after that, other features are set during an initialization. During normal VNC session client sends only information of key or pointer events and asks to send him back a graphic buffer of a server desktop (screen). Server is responsible for handling all messages from client and generates a pixel buffer enclosed desktop area.

MODIFIED REGION CODING

There may be large regions which have no change between consecutive screen images, depending on applications to be shared. Motivated by this observation, we propose a modified region coding, which encodes the modified region only as illustrated in Fig. 6. Note that modified region coding is allowed for MJPEG only, and it is not applicable to typical video coding standards such as MPEG and H.264 due to their inter-frame coding. In the first step of modified region coding, a screen image is segmented into unit rectangles which are fixed size blocks. Then, difference detection between current and previous screen images is performed for each unit rectangle. If all pixel values are identical, the unit rectangle is regarded as a skip block, which does not need to be transmitted. If any difference is detected, the unit rectangle is encoded and is transmitted to the client as usual.

Following Techniques for modified region encoding are to be used in our prototype.

- Hierarchical region detection algorithm
- Dual region detection algorithm

Protocol Improvement

RFB protocol operation is changed from serial to parallel in order to remove idle time redundancy. That modification is actually implemented into our prototype system, and the idle time is measured. As shown in Table, we can significantly reduce the redundant idle time, and it can contribute to boost screen update rate.

IV. SYSTEM IMPLEMENTION

A) HARDWARE REQUIREMENTS

1.	PROCESSOR	: PENTIUM IV
2.	SPEED	: 2.4GHZ
3.	MEMORY	: 128 MB DDR RAM
4.	HARD DISK	: 20 GB
5.	JAVA ENABLED WEB ACCESS MOBILE (GPRS)	

B) SOFTWARE REQUIREMENTS

1. LANGUAGE : JAVA (GEL EDITOR)
2. VNC SERVER
3. VNC VIEWER
4. J2ME WIRELESS TOOLKIT

V. OUTCOMES

We will extend defined protocol. Three main extensions will be there: new encodings, pseudo encodings and new security types. New encodings is added to speed up communication (for example by using better compress algorithm) if encoding is not supported by server VNC messages are simply ignored. Pseudo encodings is usedby client to inform server that is supports certain extension. If developer would like to add new protocol and even give up RFB new security types can be used. Thus we will be able to view the desktop of the computer remotely from the Symbian mobile with much faster rate due to modified region encoding.

V. CONCLUSIONS

We are proposing system for mobile VNC, and reported practical performance evaluations. To integrate video codecs into our VNC system, the existing RFB protocol is extended, preserving backward compatibility. Also, protocol operations are modified to parallel for reducing unnecessary idle time. In addition to the adoption of video codec we propose a modified region coding to further reduce the encoding time of screen images. Based on numerous experiments, we found that MJPEG is the most suitable for mobile VNC systems in terms of both complexity and compression ratio. Besides, the proposed modified region coding can further decrease encoding time, and consequently increase screen update rate at the client. Various practical and diverse experimental results demonstrate that the proposed methods guarantee fast screen image encoding without visual quality degradation. In particular, modified region detection is significantly effective for gaming video contents with small texture regions.

REFERENCES

1. Ha-Young Ko, Jae-Hyeok Lee, Jong, Ok Kim“Implementation and Evaluation of Fast Mobile VNC Systems”,Consumer Electronics, IEEE Transactions on, vol.58, 4, 2012
2. Md. Sanaullah Baig, Rajasekar M and Balaji P, “Virtual Network Computing Based Remote Desktop Access”,International Journal of Computer Science and Telecommunications, vol.3, 5, 2012
3. Adam Skurski, Bart_omiej_wiercz,” VNC-based Remote Control for Symbian OS Smartphones”, Mixed Design of Integrated Circuits & Systems, 2009. MIXDES '09. MIXDES-16th International Conference, 2009
4. C. Taylor and J. Pasquale, “Improving video performance in VNC under high latency conditions,” Proc. of the International Symposium onCollaborative Technologies and Systems (CTS 2010),2011
5. Brown, L. D., Hua, H., and Gao, C.,” A widget framework for augmented interaction in SCAPE”, Proceedings of the 16th annual ACM symposium on User interface software and technology, vol.5, 2, 2003
6. Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, vol.79, 5, 2005