



Kumaraguru College of Technology
Coimbatore

Department of Computer Applications

MOBILE VIRTUAL NETWORK COMPUTING

Bonafide Certificate

By

V.S.SHANMUGAM
Reg. No 71203621048

of

KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore
(Affiliated to Anna University)

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

*In the partial fulfillment of the requirements
for the award of the degree*

of

MASTER OF COMPUTER APPLICATIONS

June 2006

Certified that this project report titled **MOBILE VIRTUAL NETWORK COMPUTING** is the bonafide work of **Mr.V.S.Shanmugam (Reg.No.71203621048)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

[Signature]
GUIDE

[Signature]
HEAD OF THE DEPARTMENT

Submitted for the University Examination Held on 30/6/2006

[Signature]
INTERNAL EXAMINER

[Signature]
EXTERNAL EXAMINER

M Metronet Communication

No. 5, Kumaran Colony
Vadapalani
Chennai - 600092
Phone : 044-26253889

iii

MNCL/ADMIN/057/06

01/06/2006

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. V. S. Shanmugam (03MCA50), Kumaraguru College of Technology, Coimbatore has successfully completed the project titled "**MOBILE VIRTUAL NETWORK COMPUTING**" for our concern, under the guidance of Mr. S. Anand (Project Leader). His effort towards successful completion of the project from January 2006 to June 2006 is appreciable.

We wish all the best in his future endeavors.

For Metronet Communications,

[Signature]
Gayathri
Administrator

Abstract

Virtual Network Computing is, in essence, a remote display system which allow a user 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. Mobile Virtual Network Computing is an attempt to port Virtual Network Computing to mobile phones using the J2ME API.

Mobile VNC (Virtual Network Computing) it to make it possible to view and fully- interact with one computer from a mobile device anywhere in the Internet. By installing a VNC server anyone can display a desktop on device and interact with it. Virtual Network Computing requires only reliable network transport (TCP/IP) and a way displaying pixels (either writing directly to the frame buffer or going through a windowing system). The MVNC 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 MVNC viewer can access any file remotely, modify through Mobile Phones from any part of the world. To reduce the user effort and solve problems inherent to the cellular phone's small screen, several functions are provided on the cellular viewer.

The working process of the entire concept executed as follows:

The system to be accessed must be running a VNC server and it must be attached to the network. A proxy is used to send the image of the desktop to the cellular phone to convert different devices, to suppress the network traffic and support recovery from an unscheduled disconnection. Now the user must specify the IP address of the particular system and password of the server. After the checking process is done the server will send a response to the users mobile by displaying the desktop in the mobile phone.

ACKNOWLEDGEMENT

I wish to express my sincere thanks to **Dr. Joseph V.Thanikal, Principal**, Kumaraguru College of Technology, Coimbatore, for permitting me to undertake this project.

My deepest acknowledgement to **Dr. M.Gururajan**, HOD, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for his timely help and guidance throughout this project.

I am greatly indebted to my guide **Ms.V.Jalaja jayalakshmi**, Lecturer, Department of Computer Applications, Kumaraguru College of Technology, for her valuable guidance and encouragement at every stage of this project work.

I express my sincere thanks to **Ms.V.Geetha**, Assistant professor, Department of Computer Applications for her timely suggestions provided at every stage of the project.

I am very thankful to **Mr.S.Anand**, Project Lead, Metro net Communication, who rendered genuine support and timely technical help for better improvisation.

Finally I thank **my lovable parents and friends who helped** me in many ways during the course of project and have made it great success.

2.3.1 External Interface Requirements	10
2.3.1.1 Hardware Interfaces	10
2.3.1.2 Software Interfaces	11
2.3.1.3 Communication Interfaces	11
2.3.2 Performance Requirements	15
2.3.3 Design Constraints	16
2.3.4 Other Requirements	16
3. SYSTEM DESIGN	18
3.1 Decomposition and Description	18
3.1.1 Component or Module I Preliminary Description	20
3.1.2 Component or Module II Preliminary Description	21
3.1.3 Component or Module III Preliminary Description	22
3.1.4 Component or Module IV Preliminary Description	23
3.1.5 Component or Module V Preliminary Description	24
3.2 Component Detailed Design Description	25
4. SYSTEM TESTING	33
4.1 Test Environment	33
4.2 Software Testing	34
5. SYSTEM IMPLEMENTATION & MAINTENANCE	36
6. CONCLUSION AND FUTURE ENHANCEMENT	38
7. APPENDICES	39
8. REFERENCES	49

TABLE OF CONTENTS

	PAGE NO
ABSTRACT	iii
ACKNOWLEDGEMENT	iv
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS	vii
1. INTRODUCTION	1
1.1 Project Definition	1
1.2 Organization Profile	3
2. REQUIREMENT ANALYSIS AND SPECIFICATION	4
2.1 Introduction	4
2.1.1 Purpose	5
2.1.2 Scope	6
2.1.3 Project Definition	7
2.1.4 Overview	7
2.1.4.1 Current System	8
2.1.4.2 Limitations of Current System	8
2.1.4.3 Proposed System	9
2.1.4.4 Objectives of Proposed System	9
2.2 General Description	9
2.2.1 User Characteristics	9
2.2.2 Assumptions and Dependencies	10
2.3 Requirements	10

LIST OF FIGURES

FIGURE DESCRIPTION	PAGE NO
Figure 3.1 MODULE DIGRAM FOR MVNC	18
Figure 3.2 MOBILE VNC ARCHITECTURE	19
Figure 3.3 VNC CLIENT SERVER COMMUNICATION	25
Figure 3.4 VNC ARCHITECTURE	28

LIST OF ABBREVIATIONS

VNC- Virtual network computing
MVNC- Mobile Virtual Network Computing
RFB-Remote Frame Buffer
TCP/IP-Transmission Control Protocol

Chapter I

INTRODUCTION

1.1 Project definition

What exactly is MVNC?

MVNC stands for Mobile Virtual Network Computing. It is, in essence, a remote display system which allows the user to view a virtual 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.

Mobile virtual computing is an attempt to port Virtual Network Computing to mobile phones using J2ME API.

Why Virtual Network Computing?

For this simple mode of operation, a similar effect can be achieved by installing an X server on a PC. The important factors that distinguish VNC from other remote display systems such as X are as follows:

- No state is stored at the viewer. This means you can leave your desk, go to another machine, whether next door or several hundred miles away, reconnect to your desktop from there and finish the sentence you were typing. Even the cursor will be in the same place. With a PC - X server, if your PC crashes or is restarted, all the remote applications will die. With VNC they go on running.
- It is small and simple. The server and viewer are small in size and can be run directly from a floppy. There is no installation needed and so installing and maintaining VNC on a big network is simple.
- It is truly platform-independent. A desktop running on a Linux machine may be displayed on a PC or a Solaris machine or any number of other architectures. The simplicity of the protocol makes it easy to port to new platforms.

1.2 Organization Profile

About the Company

Metronet Communication was formed in 1999. In response to the following market conditions: Metronet Communication is a Product development company.

Metronet Communication has entered into Nan technology development products. Products visions are towards Wireless Networking, Embedded systems, DSP, Networking WCDM, 3G, CDMA areas. Keeping the long term growth objectives in mind, Metronet Communication evolved business relationships with the leading companies in providing software solutions. It plans to address both overseas and the Indian Market.

Metronet Communication has established relationship with clients, both in India and abroad who often contact us for developing projects. Metronet Communication keeps in touch with emerging and growing companies. Metronet Communication presents their abilities and achievements to someone in a position not only to fully appreciate them. Metronet Communication are not part of a network which allows others access to their identity / information, their identity will never be shared with any one except a potential employer. The services offered are: Software Development – Off Shore / On-Site on Java, .Net, Telecom, DSP and Embedded Systems.

- Product on DSP, Micro controller and Embedded Systems
- IT Enable Services
- Product Related Solutions
- The Consultancy Service

- A Java viewer is present, which will run in any Java-capable browser. A Windows NT server is present, allowing users to view the desktop of a remote NT machine on any of these platforms using the same viewer.
- It is sharable. One desktop can be displayed and used by several viewers at once, allowing CSCW-style applications (Computer Supported Cooperative Work).

This MVNC (Mobile Virtual Network Computing) which we are going to develop will possess the same nature of a Virtual Network Computing but the major difference is that the mobile which we are using. Here mobile phones can be used to view a system with which the user wants to interact with or with which the user likes to take the whole control of it.

Normally if we use the existing system, we can only capture the desktop of a system, which can only view in another system. But in this MVNC the user can interact with a particular system and can take the whole control of it just by using his or her mobile phone. Here the mobile phone used by the user will be able to capture a particular system with which the user wants to interact with. This above mentioned features could be achieved by developing this project in such a manner in which certain conditions to be followed. A VNC server must be installed in the particular system with which the user wants to interact with.

Thus the user can view a particular system using his or her mobile. The user can also interact with that particular system by controlling the system using a mobile phone. This includes the actions of opening a document, typing a document and doing alterations just with the help of the mobile phone. This interaction is possible wherever the user goes. No matter where the user is. I.e. the user can view the desktop of a system in his or her mobile phone from any place provided the IP address of the system and the password of the server must be known.

CHAPTER 2

REQUIREMENTS ANALYSIS AND SPECIFICATIONS

2.1 Introduction

It is remote control software that allows the user to view and interact with a computer using his or her mobile. It means that the user can view and interact with the system which is anywhere on the Internet. This is possible by acquiring the desktop of the particular computer. After capturing the desktop of that particular system on to the user's mobile he or she can take control over that. I.e. interaction can be done.

The computers need not be of a same type. It means the user can access any type of the system's desktop using his or her mobile. So, for example the user can use it to view the Linux machine as well as Windows on his or her mobile. Here the main thing is that the particular system with which the user is going to interact with must be having a VNC server installed in it. For that VNC server a password will be made set. In the client side (Mobile) the VNC viewer will be there. Now if the user wants to interact with a particular system means then the corresponding IP address of the system and the password of the VNC server must be provided. Based on this only the authentication will be done and then the desktop of that particular system will be captured. I.e. it can be viewed through the mobile and further alterations can be done after that.

While the user is controlling a system using a mobile phone as mentioned above, certain options will be provided for the user. Here the system is designed in such a manner when the user clicks the Menu in his or her mobile then he or she can view certain things like ENTER, REFRESH, OPTIONS. By selecting those items the user can do things accordingly with that system. The user will be provided with many important things, which are necessary in order to interact with a system. Thus by using the options provided the user could make the interaction effectively.

The user can interact with a particular system which is on the Internet using his or her mobile (no matter wherever the user is) i.e. the user can interact with the system from any place just with a help of a mobile.

2.1.1 Purpose

The purpose of developing this system is to create a virtual environment for users to work in their mobile phones and also to monitor user's activities of a large network and to provide remote assistance by taking control of a remote desktop.

The main purpose of developing this Mobile VNC is to allow a person at a remote device to assume control of another computer across a network, as if they were sitting in front of the other computer. The next important thing for developing MVNC is to allow the administrators to take control of employee machines to diagnose and fix problems, or to access and administer server machines without making a trip to the console. Using this Mobile VNC, troubleshooting the computer of a distant less-technically-savvy relative is possible. For the business user, MVNC can be used to provide a flexible environment by allowing employees to access their office desktop and server machines from any machine in the company's offices or from other remote sites, regardless of the type of computers involved at either end. Using this MVNC File Access, Video access, Printer Access, Scanner Access etc... can also be done.

Another important purpose for developing this MVNC is to use the same in educational contexts, for example to allow a distributed group of students simultaneously to view a computer screen being manipulated by an instructor, or to allow the instructor to take control of the students' computers to provide assistance. In a network a particular system can be selected and the same can be controlled using this MVNC through a mobile device. Thus Mobile VNC has been developed for this purpose.

4. Remote Education

VNC can also be used in educational contexts, for example to allow a distributed group of students simultaneously to view a computer screen being manipulated by an instructor, or to allow the instructor to take control of the students' computers to provide assistance.

2.1.3 Project Definition:

Mobile VNC (Virtual Network Computing) is to make it possible to view and fully-interact with one computer from a mobile device anywhere in the Internet.

2.1.4 Overview

MVNC consists of two components. A server, which runs on the computer the user want to remotely access, and a viewer, which runs on the mobile the user have. There are two important features of MVNC:

- The server and the viewer may be on different machines and on types of computer. The protocol that connects the server and viewer is simple.
- No state is stored at the viewer. Breaking the viewer's connection to the server and then reconnecting will not result in any loss of data. Because the connection can be remade from somewhere else.
- VNC being a Platform Independent tool enables users to take control of a different machine with different Operating system. So any user running VNC in Windows can take control of a Linux desktop remotely. Thus in MVNC the user can access any type of computer.

WinVNC is a VNC server that will allow the user to view the Windows desktop from any VNC viewer. Because standard versions of Windows only support a single graphical user being logged in at any one time, WinVNC makes the existing desktop of the PC available remotely, rather than creating a separate desktop as happens with the UNIX server. VNC does not make a Windows machine into a

2.1.2 Scope

The scope the project can be explained in a detailed manner with the help of some sub headings which is as follows:

1. Remote Network Administration (System Configuration)

MVNC allows a person at a remote computer to assume control of another computer across a network, as if they were sitting in front of the other computer. An equally popular business application of VNC is in remote system administration, where VNC is used to allow administrators to take control of employee machines to diagnose and fix problems, or to access and administer server machines without making a trip to the console.

2. Remote Trouble Shooting

For the individual user, one common scenario is using VNC to help troubleshoot the computer of a distant less-technically-savvy relative.

3. Remote Business Application

For the business user, VNC can be used to provide a flexible environment by allowing employees to access their office desktop and server machines from any machine in the company's offices or from other remote sites, regardless of the type of computers involved at either end.

a) File Access, Video access etc...

b) Printer Access

c) Scanner Access etc...

multi-user server in the same way that Windows Terminal Server does. Multiple users can therefore access a single Windows machine, but if they all connect at the same time they will all see the same desktop. VNC software requires a TCP/IP connection between the server and the viewer. This is the standard networking protocol on LANs, WANs, broadband and dialup ISP. Each computer has a unique IP address.

2.1.4.1 Current System

The previous system in use hangs when multiple users get connected or when multi session is established. This makes all processes to die and connection has to be established from the first for all individual clients with an individual server. Thus VNC came up which is the current system.

The System available these days enables system-to-system access. i.e. here the user can access a system, which is in a network through another system only. No mobile phones can be used here. Here the effectiveness of capturing the desktop will not be in a great extent, as mobile phones are not used. Moreover in this current system the advantage of accessing a particular system wherever the user go will not be possible.

2.1.4.2 Limitations of Current System

- No specific tool is used for connecting the mobile device.
- In the current system there is no usage of the mobile phones.
- So the effectiveness of desktop capturing will be automatically reduced.
- Thus a system can be accessed through another system but not through a mobile.

2.1.4.3 Proposed System

The proposed system is to allow the user 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

2.1.4.4 Objectives of the proposed system

- Support for Linux and Windows system, which makes it a cross platform tool so, that Linux desktop and as well as the windows can also be captured using the mobile.
- A wide range of encoding techniques to choose from for encoding and transmission of data from server to viewer.
- In servers, decreased JPEG quality factors for low quality levels. This improves Bandwidth usage while the image quality remains satisfactory in most cases. In client's JPEG compression are now enabled by default.
- Excellent User Graphical Interface.
- Session Recording is possible, which helps in viewing action done by a user at a particular time.

2.2 General Descriptions

This section describes in detail the characteristics of the users involved in the system. It also explains the dependencies of the system.

2.2.1 User Characteristics

Here the developer is the user of the system. He can use the mobile as a viewer to access a remote desktop or he can use the mobile as a server to provide information to the systems connecting to his system.

An Administrator is present who installs all the servers in the network. It is not a compulsion that the Server should be run by an Administrator. Any user can

run the program provided he enters a password, which should be known by the user who connects to the system as a client. It is the duty of the administrator to maintain all passwords installed in the Server machines and allowing proper access to the clients when they connect. In a small network an administrator will not be needed instead the users themselves assign passwords for the servers and keep the system working.

2.2.2 Assumptions and Dependencies

It is assumed that all users using the viewer know the IP address or the computer name of the server to be connected and the password assigned to the server in order to establish the connection. This is absolutely necessary for the user who intends to connect a system, which is in a network using his or her mobile.

2.3 Requirements

2.3.1 External Interface Requirements

The following describes the hardware and software interface requirements.

2.3.1.1 Hardware Interfaces

Client & Server Machines

Minimum:

- ✓ Pentium 233 Mhz
- ✓ 32 MB RAM
- ✓ Mouse
- ✓ Standard Keyboard
- ✓ 2 GB Hard Disk Drive
- ✓ Mobile Phone

Recommended:

- ✓ Pentium II 350 Mhz,
- ✓ 128 MB RAM
- ✓ Mobile (7600 or Blackberry)
- ✓ Standard Keyboard
- ✓ 8 GB Hard Disk Drive

2.3.1.2 Software Interfaces

Operating System

- ✓ Windows 9x/XP

Server & Client Machines

- ✓ Microsoft Windows 9x/Me/2000/XP
- ✓ Linux 8.0 (or Higher)

S/W Languages Used

- ✓ Java
- ✓ J2ME

2.3.1.3 Communication Interfaces

- ✓ Ethernet Card
- ✓ LAN Connection

About the software:

JAVA

Java is an Object Oriented Programming language developed at Sun Microsystems in June 1995. Java has built-in support for threads, networking and a

vast variety of other tools. The amazing thing about java is that it can be used to create a huge variety of applications and is noted for its 'Write once Run anywhere' characteristic.

Java is simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded and dynamic language.

The Java architecture consists of Java Virtual Machine (JVM), which is an abstract computer that runs compiled java programs. The JVM supports object oriented programming directly by including instructions for object method invocation. The java compiler generates architecture-independent byte codes.

Benefits of java

Java allows the user to:

- Write robust and reliable programs.
- Build an application on almost any platform, and run that application on any other supported platform without having recompiling your code.
- Distribute your applications over a network in a secure fashion.

Java has an extensive use of library of routines for copying with TCP/IP protocols like HTTP and FTP. Java application can open and access objects across net local Universal Resource Locator (URL) with the same ease as accessing the local file system. The Remote Method Invocation method enables communication between distributed objects.

Overview of J2ME:

J2ME is a platform for small devices that is intended eventually to replace the various JDK 1.1-based products with a more unified solution based on Java 2. Unlike the desktop and server worlds targeted by J2SE and J2EE, the

micro-world includes such a wide range of devices with vast different capabilities that it is not possible to create a single software product to suit all of them. Instead of being a single entity, therefore, J2ME is a collection of specifications that define a set of platforms, each of which is suitable for a subset of the total collection of consumer devices that fall within its scope. The subset of the full Java programming environment for a particular device is defined by one or more profiles, which extend the basic capabilities of a configuration. The configuration and profile or profiles that are appropriate for a device depend both on the nature of its hardware and the market to which it is targeted.

Configurations

To support the broad range of products that fit within the scope of J2ME, Sun introduced the Configuration. A Configuration defines a Java platform for a broad range of devices. A Configuration is closely tied to a Java Virtual Machine (JVM). In fact, a Configuration defines the Java language features and the core Java libraries of the JVM for that particular Configuration. The dividing line as to what a Configuration applies is for the most part based on the memory, display, network connectivity (or limitations of) and processing power available on a device.

The Sun J2ME FAQ states the following: "The J2ME technology has two design centers—things that you hold in your hand and things you plug into a wall." This may be a good general definition, but that's exactly what it is, general. Don't let this be your sole guide in deciding which Configuration applies. Following are typical characteristics of devices within the two currently defined Configurations:

Connected Device Configuration (CDC)

- 512 kilobytes (minimum) memory for running Java
- 256 kilobytes (minimum) for runtime memory allocation
- Network connectivity, possibly persistent and high bandwidth

• *Paused:* A MIDlet is placed in the paused state after the constructor has been called, but prior to being started by the application manager. Once the MIDlet has been started, it may alternate between the Paused and Active states any number of times during its lifecycle.

• *Active:* The MIDlet is running.

• *Destroyed:* The MIDlet has released any resources it acquired, and has been shut down by the application manager.

2.3.2 Performance Requirements

Response time

In any application developed, the performance time should be faster as much as possible. In this the response time after giving the IP address and the password will be of a minimal one. Similarly the connection will also take place quickly just by checking the password (i.e. the encrypted one) whether it matches with the existing one or not.

The next is the authentication process. This will also be in a quick manner as soon as the connection is made correctly with the server the desktop will be captured in a quick time.

Number of connections to the system

The main advantage of this is that many clients can access one server. More than one client can access the server but only thing is that password and the ip address must be correct.

Connected, Limited Device Configuration (CLDC)

- 128 kilobytes memory for running Java
- 32 kilobytes memory for runtime memory allocation
- Restricted user interface
- Low power, typically battery powered
- Network connectivity, typically wireless, with low bandwidth and intermittent access.

Although this division seems pretty clear, this won't always be the case. Technology is continually advancing. The technology offers more processing power, with increased memory and screen capabilities; the overlap between these categories will become larger. This is a nice segue to our next discussion, Profiles.

Profiles

It's all well and good that devices will fall within one Configuration or the other. For example, a typical cellular phone, PDA and pager will all fit the guidelines of the CLDC. However, what seems limiting to one device in a Configuration may be abundance to another. Recall the analogy of the cellular phone screen size versus that of a PDA.

MIDlet

A MIDlet is an application that is built upon the MIDlet class. The application manager communicates with a MIDlet through methods in this class. This communication is a two-way street. As an example, just as the application manager can pause a MIDlet can make a request to be paused.

MIDlet Lifecycle

A MIDlet goes through several phases as part of its lifecycle and is always considered to be in one of three states:

2.3.3 Design Constraints

Standard Compliance

The system is designed in such a way that in the password column the user must enter a word, which are not less than six characters. Then the IP address is also a must for the user in order to acquire the desktop of a particular system. The user should specifically mention the encoding type in which the user likes to acquire the desktop in order to follow that particular encoding. The system is designed such that whatever the type of encoding the user select the display will be in that form only. Moreover the system is design such that the user can only select the type of the encoding from the given menu alone. The system is developed in such a way that the computer, which the user likes to connect, must have the VNC server installed in it.

Hardware limitations

This MVNC is developed in Java and J2ME.any change in the design of this project can be done only in the systems, which support the above software's. Only certain types of mobile phones will support this. So those mobile phones must be used in order to achieve a good result.

2.3.4 Other Requirements

Security

The security advantages of MVNC

- Here in MVNC the security rate is high because of the password processing in the connection module and the authentication module.
- Normally the IP address for the systems will be unique. This MVNC will work based on the system's IP address the security rate will be high.
- It means that the exact system alone can be captured. This is possible

because of getting the password from the user.

Reliability

The reliability of this project has been proved after testing it. This project is used by the users in a proper manner by following the instructions properly and as a result of it the user can able to view the particular system to which he or she wants to interact with. The user will only be able to view the appropriate system whose IP address is given. Once if the proper IP address of the system and the server password is given by the user means then the user will be able to view that particular system only.

User Training

Training places an important role in the system implementation. Its purpose is to ensure that all the personnel who are to be associated with the system process are to be trained and necessary knowledge and skills are being developed to cope up with the system

CHAPTER 3

SYSTEM DESIGN

3.1 Decomposition and Description

MODULE DIAGRAM FOR MVNC:

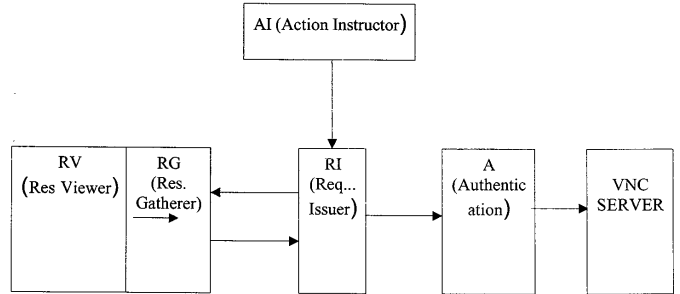


Fig 3.1 Module Diagram for MVNC

ARCHITECTURE OF MOBILE VNC

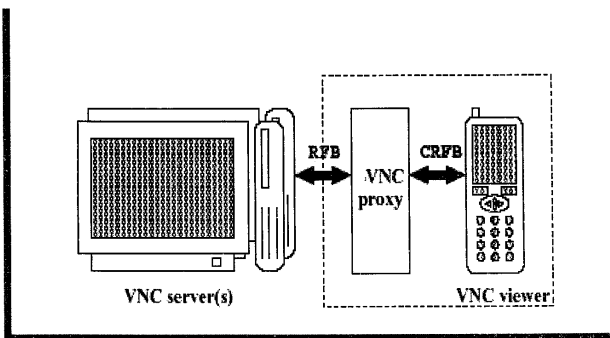


Fig: 3.2 Mobile VNC Architecture

3.1.1 Component or module I preliminary description

Authentication Module:

This module deals with the IP address of the appropriate system and the password field. Here in this module the user has to give the IP address of the system, which he or she wants to connect with the password. As soon as the IP address and the password is given by the user the proper encryption technique will be followed and then the password will be checked whether it matches with the original one or not. Only if the password field data matches with the original password of the server, the connection will be established.

This authentication module acts as the first module in this project. This module is designed in such a way that as soon as the user types the IP address and the corresponding password of the server in the corresponding places, the system must follow certain steps, which includes several encryption methods to make a connection between the client and the server. The above diagram shows the authentication module and also as shown in the figure, the flow will be maintained. I.e. only after this authentication module, the other modules will come in to play. Authentication plays a vital role in this system, which deals with the proper connection between the client and the server. Only after the proper authentication, the further things can be preceded. Once if the password does not match with the existing one means then it cannot be preceded further. I.e. if the password is incorrect means then the system will not precede to the next step instead a message will be displayed such that indicating the user to reenter the IP address or the password. Thus this is about the authentication module.

3.1.2 Component or Module II preliminary description

Request issuer:

This module comes after the authentication module. Once the IP address and the password is given by the user both will be checked and after verifying that the password matches with the original one the connection will be established as mentioned in the previous module. Now the client has to send a request to the server according to its requirement. Here this request issuer sends the request to the request gatherer, which is our next module. This request issuer will send the corresponding requests to the request gatherer in order to get a proper response from the server. So the request will be passed to the server through the response gatherer, which is our next module.

As mentioned in the above paragraph the request issuer will mainly deal with the request made by the client part. All the requests, which are made by the client part to the server in order to retrieve the proper result, will be in control of this request issuer. In order to make a communication between the client and a server, the request plays an important role. Only after receiving a request from the client, the server will start responding to the client. Based on the request only the server will respond. So the main thing in this part is that the client must make a request to the server accordingly so that the server could be able to respond to that request. Hence this request issuer will handle all the requests in order to achieve a proper communication. This request issuer will be the second module in our system, which is followed by the other three modules. Thus this request issuer plays a vital role in communication between the client and the server in order to produce a good result.

3.1.4 Component or Module IV preliminary description

Response Viewer:

This module response viewer plays a vital role in this project. Response viewer is the one, which screens all the response, which comes from the server according to the request made by the client. The response from the server will be in a first come first serving basis. All the responses from the server will reach this response viewer and then only the processing will take place. This module, which comes after the response gatherer views all the responses from the server to, the client based on the client's request.

For example if we consider the Menu part of our project, the menu part will be containing certain options, which includes Mouse mode, normal mode etc. if the client i.e. if the user request's for the mouse mode means then the appropriate request will be first gathered by the request issuer as mentioned above and this request issuer will place this request to the server and then the response from the server according to this request will be collected by the response gatherer and will reach the client through the response viewer. Thus the main part of this response viewer is to view all the responses accordingly and then that will send that to the client which places the request.

3.1.3 Component or Module III preliminary description

Response Gatherer:

This module is mainly meant for gathering all the responses, which comes from the server. These responses will come from the server according to the request send to the server through the request issuer. The main work of this response gatherer is to gather each and every response from the server based on the request from the client to the server. Here all the encoding and decoding techniques will be followed in order to receive the responses from the server to the client.

As mentioned above, the word response here points the response from the server. Normally the response will come from the server based on the request made by the client. The response from the server will be in a first come first serving basis. For example if we consider the desktop capturing means, now once the desktop is captured the server will send the whole desktop part to the client. Now as and when the changes occurring in that particular system the server must update that and send to the viewer. So this update part will be done according to the changes made in the system. I.e. if any file is opened first means then that particular action will be captured and will be updated to the client. Thus the response gatherer collects all the responses and will deliver accordingly. This response gatherer collects all the responses and will send it to the response viewer, which is the next module of our project. This response gatherer collects the responses from the server accordingly i.e. in a serial manner.

3.1.5 Component or Module V preliminary description

Action Instructor:

Here in this module all the actions will be captured and it will send through the request issuer. Request issuer in turn follows the same cycle as mentioned above in order achieves a proper communication between the client and the server. Once the request is passed to the request issuer, it will take that to the server side and then the server will react according to the request made. After the acknowledgement and the reaction of the server the response gatherer will come in to play.

As mentioned earlier the response gatherer will take complete responsibility in order to gather all those requests and also it will pass those requests to the response viewer through which the response will reach the client. All this process will take place in a sequential manner as shown in the above diagram. All this above-mentioned process will take place only if this action instructor works properly. It means that the action instructor plays a vital role in giving instruction to each and every module so that the response from the server will reach the client in a correct manner. Thus by following all the above-mentioned steps proper communication can be achieved so that the user requirement (capturing the remote desktop using a mobile phone) can be fulfilled.

3.2 Component Detailed Design Description

Technology behind VNC

The RFB Protocol

Introduction

RFB ("remote frame buffer") is a simple protocol for remote access to graphical user interfaces. Because it works at the frame buffer level it is applicable to all windowing systems and applications, including X11, Windows 9x/NT and Macintosh.

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 frame buffer originate (i.e. the

Windowing system and applications) is known as the RFB server.

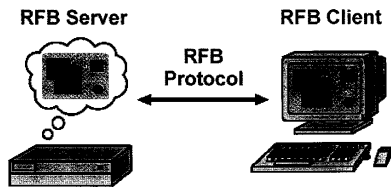


Fig: 3.3 VNC Client Server Communications

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.



that the server must always be able to supply pixel data in the form the client wants. However if the client is able to cope equally with several different formats or encoding, it may choose one, which is easier for the server to produce.

Pixel *format* refers to the representation of individual colours by pixel values. The most common pixel formats are 24-bit or 16-bit "true colour", where bit-fields within the pixel value translate directly to red, green and blue intensities and 8-bit "colour map" where an arbitrary mapping can be used to translate from pixel values to the RGB intensities.

Encoding refers to how a rectangle of pixel data will be sent on the wire. Every rectangle of pixel data is prefixed by a header giving the X,Y position of the rectangle on the screen, the width and height of the rectangle, and an *encoding type* which specifies the encoding of the pixel data. The data itself then follows using the specified encoding.

The protocol can be extended by adding new encoding types. The encoding types defined at present are *raw* encoding, *copy rectangle* encoding, *RRE* (rise-and-run-length) encoding, *CoRRE* (Compact RRE) encoding and *hextile* encoding. In practice we normally use only the *hextile* and *copy rectangle* encoding since they provide the best compression for typical desktops.

VNC Architecture

The Architecture of VNC is based mainly on the VNC Protocol called RFB Protocol. This commonly used display Protocol is used in building VncServer. The client invokes the functions of the RFB Protocol through the VncServer.

Display Protocol

The display side of the protocol is based around a single graphics primitive: "put a rectangle of pixel data at a given x,y position". This method along with encoding for the pixel data gives a large degree of flexibility in how to trade off various parameters such as network bandwidth, client drawing speed and server processing speed for many user interface components.

A sequence of these rectangles makes a *frame buffer update* (or simply *update*). An update represents a change from one valid frame buffer state to another, so in some ways is similar to a frame of video. 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 frame buffer tend to happen soon after one another. With a slow client and network, transient states of the frame buffer can be ignored, resulting in less network traffic and less drawing for the client.

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 synthesized from other non-standard I/O devices.

Representation of pixel data

Initial interaction between the RFB client and server involves a negotiation of the format and encoding with which pixel data will be sent. This negotiation has been designed to make the job of the client as easy as possible. The bottom line is

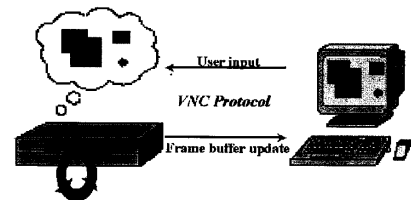


Fig: 3.4 VNC ARCHITECTURE

INITIAL HANDSHAKING MESSAGES

Protocol Version

Handshaking begins by the server sending the client a *Protocol Version* message. This lets the client know which is the latest 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 Server). It is intended that both clients and servers may provide some level of backwards compatibility by this mechanism. Servers in particular should attempt to provide backwards compatibility, and even forwards compatibility to some extent. For example if a client demands version 3.1 of the protocol, a 3.0 server can probably assume that by ignoring requests for encoding types it doesn't understand, everything will still work OK. This will probably not be the case for changes in the major version number.

The *Protocol Version* 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.

No. of bytes Value

12 "RFB 003.003\n" (hex 52 46 42 20 30 30 33 2e 30 30 33 0a)

Authentication

Once the protocol version has been decided, the server then sends a word indicating the authentication scheme to be used on the connection

No. of bytes	Type [Value]	Description
4	CARD32	<i>authentication scheme:</i>
	0	<i>connection failed</i>
	1	<i>no authentication</i>
	2	<i>VNC authentication</i>

This is followed by data specific to the *authentication-scheme*:

- **Connection failed** - for some reason the connection failed (e.g. the server cannot support the desired protocol version). This is followed by a string describing the reason (where a string is specified as a length followed by that many ASCII characters)

No. of bytes	Type [Value]	Description
16	CARD8	<i>response</i>

The server sends a word to inform the client whether authentication was successful. If so, the protocol continues with the *ClientInitialisation* message; if not the server closes the connection:

No. Of bytes	Type [Value]	Description
4	CARD32	<i>status:</i>
	0	<i>OK</i>
	1	<i>failed</i>
	2	<i>too much</i>

Client Initialization

Once the client and server are sure that they're happy to talk to one another, the client sends an Initialization message:

No. of bytes	Type [Value]	Description
4	CARD32	<i>reason-length</i>
Reason-length	CARD8	<i>array reason-string</i>

The server closes the connection after sending the *reason-string*.

- **No authentication** - no authentication is needed. The protocol continues with the

ClientInitialisation message.

- **VNC authentication** - VNC authentication is to be used. This is followed by a random 16-byte challenge:

No. of bytes	Type [Value]	Description
16	CARD8	<i>challenge</i>

The client encrypts the challenge with DES, using a password supplied by the user as the key, and sends the resulting 16-byte response:

No. Of bytes	Type [Value]	Description
1	CARD8	<i>shared-flag</i>

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.

Server Initialization

After receiving the *ClientInitialisation* message, the server sends a *Server Initialization* message. This tells the client the width and height of the server's frame buffer, its pixel format and the name associated with the desktop:

No. of bytes	Type [Value]	Description
2	CARD16	<i>frame buffer-width</i>
2	CARD16	<i>frame buffer-height</i>
16	PIXEL_FORMAT	<i>server-pixel-format</i>
4	CARD32	<i>name-length</i>
<i>Name-length</i>	CARD8	<i>array name-string</i>

Server Initialization

CHAPTER 4

SYSTEM TESTING

4.1 Test Environment

Software testing is the last phase of the software development phase. Testing contains the total development efforts.

Testing:

Testing is a process of executing a program with the intent of finding an error. This is the major part of any software development process. It has to be done in a manner that no errors should be found. The system has to be tested for all kinds of inputs and the worst cases, with an intention of finding errors. This software has been tested using the following two test techniques of software testing.

White-Box Testing:

This is also called as glass-box testing. It is a test case designed methods that uses the control structures of the procedural design to derive the test cases. Using white box testing one can derive the test cases that

- Guarantee that all the independent parts with in a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and with in the operational bounds.
- Exercise internal data stretches to assure and validate.

Black-Box Testing:

This is another way of testing a system. A black box testing examines some fundamental aspects of a system with little regard for internal logic structure of the software. Further it is used to demonstrate that software functions

Integration Testing

This is nothing but linking of different modules. As the individual modules have to be integrated, thorough checking has to be done on integration system also.

In MVNC the five modules are being integrated and is got tested to check whether it works properly or not as a whole. The testing was successful and it was found that proper response was obtained when there is a request from one module to another module.

Unit Testing

It has the goal of discovering errors in the individual module of the system. It is eased by the system structured of small, loosely coupled modules. It should be as exhaustive as possible to ensure that each representative case is handled by and that each module has been tested.

In MVNC the Unit Testing was being performed. Here each module is being tested deeply to find even some minute errors present in it and is got corrected respectively.

Acceptance Testing

It involves planning under execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements. It also performs stress tests to determine the limitations of the system.

In MVNC the Acceptance testing is being performed to check whether the implemented system satisfies its requirements and to determine the limitations of the system.

are operational, that the input is properly accepted and the output is correctly produced and that the integrity of the external information is maintained. It attempts to find errors in the following category:

- Incorrect or missing functions.
- Interface Errors.
- Errors in data structures or external data base access.
- Performance Errors.
- Initialization and termination errors.

4.2 Software Testing

Module Testing

The module testing is the verification of a single program module, usually in an isolated environment. The software is developed using modular approach and hence each module has to be tested individually and ensured not to have any errors and are up to the specifications.

In MVNC the five modules namely

- Authentication Module
- Request Issuer
- Response Gatherer
- Response Viewer
- Action Instructor

Are being tested individually in an isolated environment and it proved to be successful.

CHAPTER 5

SYSTEM IMPLEMENTATION

Introduction:

System implementation is the stage of the project where the theoretical design is tuned in to a working system. It is the important stage of the project. A lot of planning has to go for successful implementation of the system. The main stages of implementation are as follows:

Implementation planning

This is the first task in system implementation. The methods and the time scale to be adopted are decided. Once the planning is over, the major effort in computer department is to ensure that programs in the system work properly.

Training

Training places an important role in the system implementation. Its purpose is to ensure that all the personnel who are to be associated with the system process are too be trained and necessary knowledge and skills are being developed to cope up with the system.

Change Over

Change over from old system to new system may occur when:

- The system has been proved satisfactory and implementation has been completed.
- The users are satisfied with the results of the system test, training and reference manuals.
- The operation manager is satisfied with the performance of the system.

Amendment

Unless each amendment is properly document in appropriate files, subsequent amendments may be made which do not take in to account the effects of previous amendments. Without effective control procedures the system and its documentation will deteriorate.

System Audit

The system audit is an investigation to review the performance of the operating system. This verifies whether the stated objectives of the system are still valid in the present environment and also evaluates the achievements of these objectives.

The major steps that are carried and the events that encountered in these stages are summarized below:

- Training was given to the users of the system both theoretically and practically. The users were briefed on the lines of the objectives of the system, how to operate and the benefits that would reap from the system.
- The system was tested in the user's environment and the user was promoted to give his/her suggestions. It was found that most of the users responded favorably to the system.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

The System has been designed to satisfy the needs of Commercial Organizations and Educational Institutions. All the drawbacks of the Existing system are considered in developing the new system. Thus all the legacy systems, which had no interaction with each other, are removed. The new system handles client in a much efficient way. It maintains the client-server connection well. The server name is used in all places where the server IP is to be specified.

Session recording is introduced which helps the administrator or a user to save a particular session of activities and view it later. Different encoding techniques are included and the user can select his type of encoding to be done in the connection to be established based on the type of network he uses. The user of his/her choice can do compression of images explicitly.

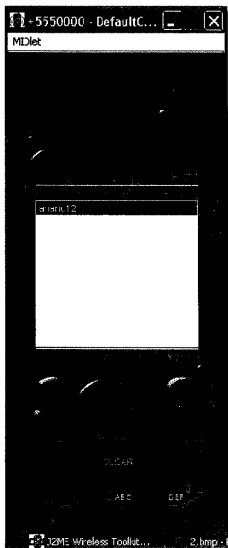
The new system developed is very flexible in nature and so any changes to be made can easily do to the system without re-designing the strategies. Also any new module can be added to the newly developed system as the system was developed with preferences for future enhancements, which includes audio transmission.

Future enhancements:

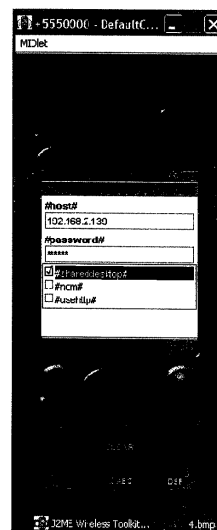
- The main thing that has to be dealt with is to use this software in any mobile devices.
- Now at present the mobiles with high configuration can use this, but in the future it has to be modified so that all the mobiles can use this.
- The compression techniques can be concentrated and can be made very effective i.e. very effective compression techniques can be used.
- Making it possible in lower bandwidths.

APPENDICES

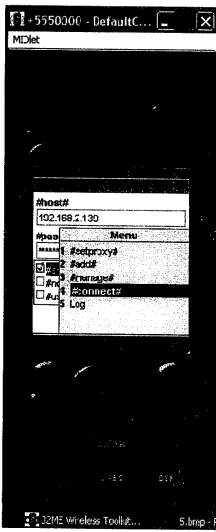
APPENDIX -1



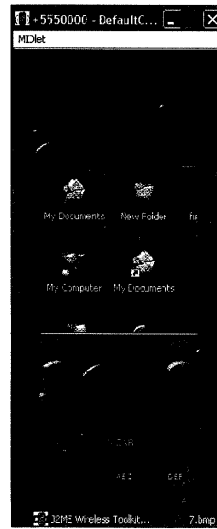
MVNC Launch



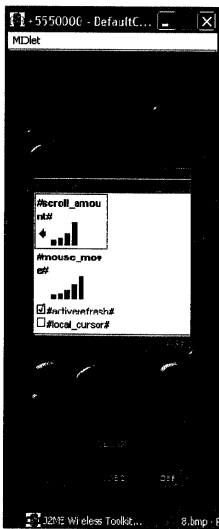
Authentication



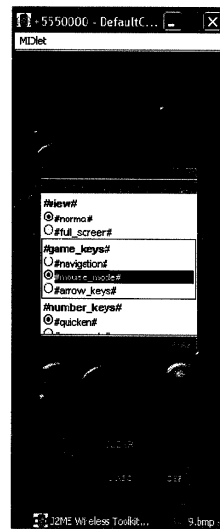
Connection Establishment



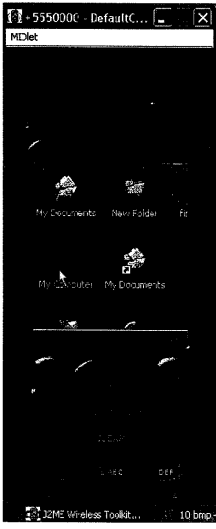
Desktop



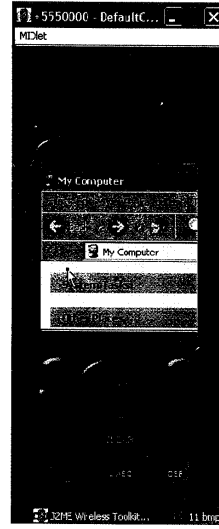
Adjustments



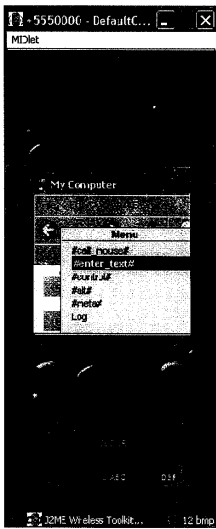
Mode Selection



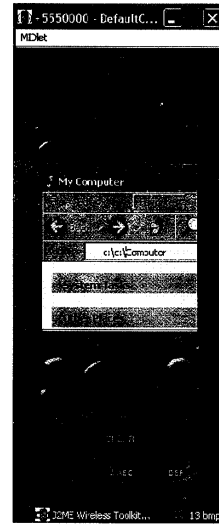
Desktop Access



Window Access



Menu Options



Text Entry

REFERENCES

- [1] Graeth M.De Bruyn, Robert Lyfareff, Ken Kroes (2001), 'Advanced J2ME Programming'
- [2] Graeth M.De Bruyn, Robert Lyfareff (1998), 'Introduction to Java', Revised & Expanded Edition.
- [3] Herbert Schildt, 'Java 2 Complete Reference', Tata McGraw-Hill Edition, Fifth Edition

WEBSITES

<http://UK-research.com/vnc>

www.realvnc.com