City University London

School of Engineering
and Mathematical Sciences

Development of Virtual Network Computing (VNC) Environment for Networking and Enhancing User Experience

Dana Mohammed Al-Malki

Supervisor: Dr S.H.Khan

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Information Engineering

August 2006

Abstract

Virtual Network Computing (VNC) is a thin client developed by Real VNC Ltd, Formerly of Olivetti Research Ltd/AT&T labs Cambridge and can be used as a collaborative environment, therefore it has been chosen as the basis of this research study. The purpose of this thesis is to investigate and develop a VNC based environment over the network and to improve the users' Quality of Experience (QoE) of using VNC between networked groups by the incorporation of videoconferencing with VNC and enhancing QoE in Mobile environments where the network status is far from ideal and is prone to disconnection.

This thesis investigates the operation of VNC in different environments and scenarios such as wireless environments by investigating user and device mobility and ways to sustain their seamless connection when in motion. As part of the study I also researched all groups that implement VNC like universities, research groups and laboratories and virtual laboratories. In addition to that I identified the successful features and security measures in VNC in order to create a secure environment. This was achieved by pinpointing the points of strength and weakness in VNC as opposed to popular thin clients and remote control applications and analysing VNC according to conforming to several security measures.

Furthermore, it is reasonable to say that the success of any scheme that attempts to deliver desirable levels of Quality of Service (QoS) of an effective application for the future Internet must be based, not only on the progress of technology, but on users' requirements. For instance, a collaborative environment has not yet reached the desired expectation of its users since it is not capable of handling any unexpected events which can result from a sudden disconnection of a nomadic user engaged in an ongoing collaborative session; this is consequently associated with breaking the social dynamics of the group collaborating in the session. Therefore, I have concluded that knowing the social dynamics of application's users as a group and their requirements and expectations of a successful experience can lead an application designer to exploit technology to autonomously support the initiating and maintaining of social interaction. Moreover, I was able to successfully develop a VNC based environment for networked groups that facilitates the administration of different remote VNC sessions. In addition to a prototype that uses videoconferencing in parallel to VNC to provide a better user's QoE of VNC. The last part of the thesis was concerned with designing a framework to improve and assess QoE of all users in a collaborative environment where it can be especially applied in the presence of nomadic clients with their much frequent disconnections. I have designed a conceptual algorithm called Improved Collaborative Quality of Experience (IC-QoE), an algorithm that aims to eliminate frustration and improve QoE of users in a collaborative session in the case of disconnections and examined its use and benefits in real world scenarios such as research teams and implemented a prototype to present the concepts of this algorithm. Finally, I have designed a framework to suggest ways to evaluate this algorithm.

Acknowledgement

To begin with, I would like to offer my thanks and gratitude to *Dr S H Khan*, who has supervised, guided, advised and had faith in me and who has been a good mentor, counsellor and friend throughout the past years.

I would also like to thank my *parents* and *siblin*gs who encouraged, motivated and had faith in me, my mother for offering me encouragement and not giving up on me, my father for tolerating, with all the support he has, my long absence to pursuit my studies, my *brother* and *sisters* who were my solid rock.

I would like to add special thanks to my sister *Amal* for always being there and for her continued patience and understanding through the years where with her support things got easier to handle.

I also would like to mention *Dr Veselin Rakocevic* for his valuable guidance in the last year of the thesis and his helpful comments and suggestions.

I would like to thank the cultural attaché of Qatar, and especially His highness Mr Nasser Al-Khalifa, who was Qatar's ambassador in London through most of the period of my studies and is currently Qatar's ambassador in Washington, for believing in my capabilities and giving me the chance to prove them.

Finally, I would like to thank the RealVNC mailing list for their help.

Declaration

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgements

Abbreviations

2G:	Second Generation Wireless
3G:	Third Generation Wireless
AMPS:	Advanced Mobile Phone System
ASP:	Application Service Provision
ATM:	Asynchronous Transfer Mode
AWT:	Java Abstract Windowing Toolkit
BER:	Bit Error Rate
BS:	Base Station
CAD:	Computer Aided Design
CC:	Cellular Communication
CDMA:	Code division multiple access
CDPD:	Cellular Digital Packet Data
CE:	Collaborative environment
CN:	Correspondent Node
CoA:	Care-of-Address
CSCW:	Computer Supported Cooperative Work
DTP:	Digital Publishing
DUPACK:	Duplicate Acknowledgments
FA:	Foreign Agent
FEC:	Forward Error Correction
GNOME:	GNU Object Model Environment
GUI:	Graphical User Interface
GSM:	Global System for Mobile Communications
GPRS:	General Packet Radio Service
HA:	Home Agent
HCI:	Human-Computer Interaction

HTML:	Hypertext Mark-up Language
HTTP:	Hypertext Transfer Protocol
IC-QoE:	Improved Collaborative Environment
IP:	Internet Protocol
ISDN:	Integrated Services Digital Network
IWF:	Interworking Function
KDE:	K Desktop Environment
LCD:	liquid crystal display
LDAP:	Lightweight Directory Access Protocol
LLC:	Logical Link Control
MAC:	Media Access Control
MAHO:	Mobile-assisted Handover
MDAC:	Microsoft Data Access Components
MN:	Mobile Node
MSC:	Mobile Switching centre
MTSO:	Mobile Telephone Switching Office
NACK:	Negative Acknowledgment
NCHO:	Network-controlled Handover
NDS:	Novell Directory Services
PCS:	Personal Communication Services
PDA:	Personal Digital Assistant
PPP:	Point-to-Point Protocol
QoS:	Quality of Service
QoE:	Quality of Experience
QoT:	Quality of Transmission
RDP:	Remote Desktop Protocol
RFB:	Remote Frame Buffer
RLP:	Radio Link Protocol
RSSHT:	Relative Signal Strength Hysteresis and Threshold

RTO:	Retransmission Timeout
RTT:	Round trip time
SACK:	Selective Acknowledgment
SLIP:	Serial Line Internet Protocol
SNR:	Signal-to-Noise Ratio
TCO:	Total Cost of Ownership
TDMA:	Time division multiple access
TCP/IP:	Transmission Control Protocol/Internet Protocol
UMTS:	Universal Mobile Telecommunications System
VNC:	Virtual Network Computing
W-LAN:	Wireless Local Area Network
W-WAN:	Wireless Wide Area Network
WYSIWIS:	What You See Is What I See
ZWA:	Zero Window Advertisement
ZWP:	Zero Window Probe

List of Figures

Table of Contents

Appendices (On CD attached)

1 Chapter One

Introduction

1.1 Motivation and Benefits

Progress in business requirements and the change in social trends require today's businesses and multi-disciplinary organisation users to use the web for a large amount of their computing functions. To meet these demands many applications despite their nature, platform-orientation and high necessity must be web-enabled to reach the growing number of remote users and to address the web computing needs of their customers. Furthermore, the advances in computers, technologies and telecommunications are encouraging many innovative ways to benefit from the ubiquity of the Internet and the web-enabled applications. One example is telecommuting where telecommuting [2] is the practice of an employee performing his or her normal duties from a remote location, typically home, on a full-or part time basis. Advances in computers and technologies permit people to carry huge computer power and large quantities of information with them and the advances of telecommunications permit transmission of other information from remote computers around the world in a matter of minutes. According to [2] remote access benefits employees at remote sites, mobile employees, telecommuters, contractors, alliance partners, customers, suppliers and channel partners. Also, remote access computing helps business comply with clean air act by reducing travel to work, family leave act where it allows employees to spend time away from the office to be with a newborn child or for family medical emergencies and service technicians who are on the road and need office connectivity to get assignments and order parts and send billing information.

Using computer technology to support groups in organisations is growing in response to the wide geographic distribution of much research and development and using temporary workers or professional service firms. Therefore Collaborative Environments (CE) are another example that benefits from the ubiquity of the Internet and can offer new opportunities for communication and collaboration, influence teaching and learning, and significantly improve the way research groups are brought together to share scientific data and ideas.

An innovative approach to facilitate using applications over the Internet and delivering business-critical applications to end-user devices is Server-based Computing, also known as thin-client technology whereby an application's logic is executed on the server and only the user interface is transmitted across a network to the client. There are two approaches to thin client computing: Web-based approach and graphics pipeline interceptions. Web based approach uses HTTP to negotiate the transfer of HTML data between the client Web browser and the Web server then the Web browser renders the HTML onto the client frame buffer. The HTTP/HTML works on ``pull-only'' data transfer methodology where such applications are prevented from generating events and thus cannot provide a rich user experience. To address this problem many solutions were designed. One of them is using Java applets to send entire applications over HTTP. However, Java applets raise numerous security concerns because HTTP is used to transport executable code to the client. Although the byte codes transmitted across the network are in compiled form, Java decompilers are readily available that will allow any user to have access to the source code of the application. In addition, the use of Java applets typically violates the thin-client principle of not running any application logic on the client. The second approach in delivering applications in a thin client environment is intercepting rendering commands sent to the graphics pipeline. A way of implementing this is to create a virtual frame buffer in the RAM of the server on which the application can render its GUI and then transporting the resulting raster image to the client. [1]

Virtual Network Computing (VNC) is a powerful tool developed by Real VNC Ltd, Formerly of Olivetti Research Ltd/AT &T labs Cambridge and had been recognised by a massive crowd around the world. VNC has been chosen as the basis of this research study because of several reasons. Firstly it is a thin client where it implements the graphics pipeline interceptions and therefore will provide a free reliable cost effective environment solutions to web-enable UNIX and Windows applications and will turn them to cross-platform web services accessed within a network via the Internet where applications on a certain platform can be viewed and used on other platforms over the Internet either by using a VNC standard client or using a web browser. Also, it will help collaboration between users without requiring any re-writing of applications which will make them collaboration-unaware and will turn single user environments to multi-user environments. Furthermore, VNC provides transparency of access when used as a remote access application.

VNC is available for everyone because of its nature of being open source and is distributed under the GNU General Public Licence as published by the Free Software Foundation. This gives it the power and popularity it reached since the day of its release to the public. The spread of using VNC will result in improvements in many areas such as education, engineering, working teams in different industrial sectors. First of all, VNC can be used in distance learning and virtual classrooms and also can be used in many educational classes where the tutor can show his/her screen to all of the students while they are sitting in front of their PCs or to allow the tutor to monitor all of the students in his class. Secondly, VNC can help and facilitate concurrent engineering where design teams and research groups cannot physically meet at the same location. Thirdly, through its thin-structure VNC will enable professionals to use applications not on their local workstations and can use different applications on different platforms and heterogeneous networks and may need to have access to remote and state of the art equipments at distance to create virtual laboratories.

It is a fundamental for one seeking the analysis, design and development of an effective application or environment to consider its users. Moreover, traditional quality of service metrics such as response time and delay no longer suffice to fully describe quality of service as perceived by users. The success of any scheme that attempts to deliver desirable levels of QoS for the future Internet must be based, not only on the progress of technology, but on users' requirements. This introduces the term Quality of experience, which is a concept comprising all elements of a user’s perception of the network and performance relative to expectation. Since VNC is a typical example of a thin client solution that can be used in distributed collaborative environments, it is interesting to investigate the QoE in VNC-based systems.

For instance, a collaborative environment has not yet reached the desired expectation of its users since it is not capable of handling any unexpected exception which can result from a sudden disconnection of a nomadic user engaged in an ongoing collaborative session. This is consequently associated with breaking the social dynamics of the group collaborating in the session.

1.2 Aims and Objectives

Generally for designing and analysing collaborative environments there are two research approaches of empirical studies and prototype development. Empirical studies investigate how groups actually work with or without computers and rely upon both quantitative and qualitative empirical data collection and analysis methodologies derived from psychology, sociology, anthropology, and other social sciences. Whereas, prototype development approach focuses on improving the state of the art in applications or identifying applications that serve new functions in supporting groups.

This thesis investigates Virtual Network Computing as a full functional featured system, thin client and a collaborative environment to support groups. It also presents a framework to improve the user's quality of experience (QoE) using VNC in general real world scenarios as well as using VNC as a collaborative environment in mobile environments in specific by adopting an intermediate approach between the empirical studies and prototype development.

Five major steps were needed to develop a written resource for VNC users ranging from novice users to system administrators and from one-time users to a heavy user and enhance its QoE: develop a full understanding of the way VNC runs and operates in different environments and scenarios, identify QoE and ways to improve it in general and in collaborative environments, identify the successful features and security measures in thin client applications, develop and implement VNC based interaction for networked groups, develop and implement a design framework to improve and asses QoE in collaborative environments.

1.2.1 Develop a full understanding of the way VNC runs and operates in different environments and scenarios

- By highlighting the TCP/IP protocol stack which is responsible of reliable data delivery of Internet applications over the network and looking at mobile computing in nomadic and wireless environments and discussed Mobile IP to address the problem with higher layer protocols like TCP which relies on IP addresses and could not deal with IP mobility without disconnecting the session to make communicating hosts unaware of the underlying movement between the subnets. I also discussed TCP performance in both wired and wireless environments and mentioned currently developed improvements to the TCP improvements such as snoop-TCP, Indirect-TCP, Mobile-TCP and Freeze-TCP

- By pinpointing the technologies that inspired the development of VNC and investigated user and device mobility and introduced thin client and mentioned its advantages and disadvantages
- By testing and implementing VNC on both UNIX and Windows platforms
- By researching all areas and settings in which VNC is used

1.2.2 Identify QoE and ways to improve it in general and in collaborative environments in particular

- By researching collaborative environments and groupware that support the work in groups and by discussing the barriers of adopting collaborative environments and the social challenges the developers of groupware face
- By discussing how to deal with delay in collaborative environments and the types of delay that mostly affect the CE
- Identifying QoE and all elements of user's perception of the network and performance relative to expectations

1.2.3 Identify the successful features and security measures in thin client applications

such as VNC and specify those which need improvement to support groups in a collaborative manner and derive the features that VNC lack after looking closely at different systems similar to VNC. These evaluations feed back into the redesign of the computer application to fit into the real-life collaborative scenario instead of starting from scratch and wasting time, money and effort with no guarantees of success deployment when used in real environments.

- By investigating VNC as a collaborative environment and its QoE and presented it in computer supported cooperative work and analysing it according to many CSCW terms
- By surveying different types of thin client systems and remote control solutions
- By investigating all security aspects of using VNC and suggesting a secure implementation on both PCs and Workstations

1.2.4 Develop and implement VNC based interaction for networked groups

- I designed four prototypes for collaboration to enable for example, a supervisor to view and monitor his research students simultaneously and enable him to compare results and monitor screens between two users who are working on the same or similar project. The first prototype is an HTML application interface that divides the screen vertically into two sessions and the second prototype, is also an HTML application that divides the screen horizontally and it contains as many VNC sessions as a user would like. I also proposed and implemented security features for the first and second prototypes by using Secure Shell encryption. The third prototype I have developed overcomes drawbacks in the first prototype and it implements scaling using an ActiveX control already developed by Thong Nguyen called VNCX. The fourth prototype is a Java application that can be used to view several VNC sessions
- I proposed using videoconferencing in a one-to-one scenario to enhance the functionality of using VNC between a supervisor and his research student and also to enhance collaboration between two research students such as engineers working on the same project and in need of discussing many aspects of the design and implementation of their project and I proposed using NetMeeting and developed a prototype to add its functionality to a VNC Interface

1.2.5 Develop and implement a design framework to improve and asses QoE in collaborative environments

- I analysed important factors that affect the outcome of a user's quality of experience in interacting with collaborative environments, which not only consist Human-to-Computer elements of interaction but concentrates mainly on Human-to-Human social interaction in the form of being a member of a group
- I outlined some technical concepts that can be applied or studied in order for developers to design or redesign and implement enhancements to current applications and to use wireless TCP to ensure high network throughput and mobile IP to allow the user's device to connect and continue to roam without dropping the connection or having to regain permission to use the network
- I proposed an improved collaborative QoE algorithm and applied it to a thin client such as VNC in the case of disconnections to eliminate frustration but before that I pinpointed what factors can affect this user Experience in the case of disconnection

1.3 Structure of the thesis

This thesis starts by highlighting in Chapter Two, firstly the TCP/IP protocol. Also, I looked at mobile computing and nomadic and wireless environments then discussed Mobile IP to address the problem with higher layer protocols like TCP which relies on IP addresses and could not deal with IP mobility without disconnecting the session to make communicating hosts unaware of the underlying movement between the subnets. And I discussed TCP performance in both wired and wireless environments and already developed improvements to the TCP improvements.

In chapter Three, firstly I pinpointed the technologies that inspired the development of VNC and introduced thin client computing where all the processing is done on the server and only the mouse movements and key events are sent to the client and which is based on the client/server paradigm. VNC is the conceptual extension of teleporting to introduce teleporting to the wider arena of the Internet therefore I looked at teleporting which was also developed by AT&T and it is based on the X windows system which is also mentioned. Another system I looked at is the video tile which is another inspiration for the development of VNC and VNC is the software version of it. Also I looked at the structure of the VNC system for UNIX and the RFB (Remote Frame Buffer) protocol which is the essence of the VNC system. Secondly, I researched collaborative environments and groupware that support the work in groups, the barriers of adopting collaborative environments and the social challenges the developers of groupware go through and discussed how to deal with delay in collaborative environments. I also researched a new term Quality of Experience (QoE) that is receiving increased interest the research community and investigated VNC as a collaborative environment and its QoE.

In Chapter Four, Firstly I tested the use of VNC on both UNIX and Windows platforms. In this chapter I presented the different issues in the installation of VNC on Windows and in using the VNC viewer on Windows. Then I presented the UNIX version on Solaris where I faced many problems in having a VNC session running on UNIX and I discussed the problems and the solutions to overcome them in the Chapter and I explained the use of the VNC viewer on UNIX. Secondly, I surveyed different types of thin client systems and remote control applications; However, there are many thin clients and remote control software therefore I choose only four of the popular software available to investigate their functionalities and their points of strength and compared and contrasted them to VNC to deduce the weak and strong points of VNC. The thin clients are Citrix's Met frame, Gryphon's Go-Global, Microsoft terminal server and SCO's tarantella. Whereas the remote control applications are NetSupport Manager, Altiris Carbon Copy, Unicenter Remote control and PCAnywhere.

Chapter Five explains all security aspects of using VNC and suggests a secure implementation on both PCs and Workstations. Firstly, I discussed and analysed VNC from the point of conforming to security measures like authentication, access control, encryption, physical security and auditing. Secondly, I researched and tested some of the security risks on VNC such as password-related risks and network-related risks. In the fourth Section I proposed security precautions on Windows. And then I implemented encryption by using SSH (Secure Shell) on both Windows and UNIX and explained it step by step and described using tunnelling (port forwarding) and after that I presented different methods that can be used with VNC for encryption like VPN, SSL used in Stunnel, Zebedee based on Blowfish encryption. In the last part I explained how to traverse firewalls to connect to a VNC system behind a firewall, which I tested between Home machines resident in London and also an overseas machine in Qatar to connect to a UNIX and Windows machines behind a firewall by tunnelling and reverse connection.

In Chapter Six, I researched all areas where VNC is used in universities, research groups and laboratories for remote access and computer support at help desks and in computer training. Another area is using it in virtual laboratories for hardware control and for software provision and in application service provision. I also mentioned two systems in which VNC is used to control home appliances and also using VNC in the field of handheld devices and mobile phones. I also designed four collaboration prototypes to enable for example, a supervisor to view and monitor his research students simultaneously and enable him to compare results and monitor screens between two users who are working on the same or similar project. I also proposed using videoconferencing to enhance the functionality of using VNC between research groups.

In chapter Seven, I analysed important factors that affect the outcome of a user's quality of experience in interacting with collaborative environments and outlined some technical concepts that can be applied or studied in order for developers to design or redesign and implement enhancements to current applications. Also, I proposed an improved collaborative QoE algorithm and applied it to a thin client such as VNC in the case of disconnections to eliminate frustration but before that I pinpointed what factors can affect this user Experience in the case of disconnection.

Chapter Eight concludes the thesis and discusses possible future work.

2 Chapter Two

TCP/IP and Mobile Environments

2.1 The TCP/IP protocol Suite

Many popular Internet applications including World-Wide Web (WWW), File Transfer Protocol (FTP) and email require reliable data delivery over the network. All modern networks are designed using a layered approach, where each layer presents a predefined interface to the layer above it and below it.

The TCP/IP is the suite of communications protocols used to connect hosts on the Internet which allows communication across multiple heterogeneous networks. TCP/IP was originally designed to allow different types of computer systems to communicate as if they were the same system.

TCP/IP Suite	
Application Layer	HTTP,SMTP,FTP,SSH,IRC,SNMP …
Transport Layer	TCP, UDP, SCTP, RTP, DCCP …
Internet Layer	IPv4 , IPv6, ARC, ICMP …
Network Interface Layer	Ethernet , 802.11 ,WiFi, Token Ring, FDDI …

Figure 2.1 TCP/IP Protocol Suite

TCP/IP is composed of four layers as shown in Figure 2.1, the network interface layer, the Internet layer, the transport layer, and the application layer. It is important to notice that this protocol runs independently of the data-link and physical layer. At these layers, the TCP/IP protocol can run on Ethernet, Token Ring, FDDI, serial lines, X.25, and so forth.

2.1.1 Layers of TCP/IP Suite

2.1.1.1 The Network Interface layer

A network interface layer also called link-layer protocol defines the network hardware and device drivers and is responsible for communicating with the actual network hardware (e.g., the Ethernet card). It hands the data it receives off the network to the Internet layer; and puts the data it receives from the Internet layer on the network wire. This is where device drivers for different interfaces reside.

The design of TCP/IP hides the function of this layer from users—it is concerned with getting data across a specific type of physical network (such as Ethernet, Token Ring, etc.). This design reduces the need to rewrite higher levels of a TCP/IP stack when new physical network technologies are introduced (such as ATM and Frame Relay).

The functions performed at this level include encapsulating the IP datagrams into *frames* that are transmitted by the network. It also maps the IP addresses to the physical addresses used by the network. One of the strengths of TCP/IP is its addressing scheme, which uniquely identifies every computer on the network. This IP address must be converted into whatever address is appropriate for the physical network over which the datagram is transmitted.

The TCP/IP protocol suite is structured around the Internet Protocol (IP) and assumes that it will be layered on top of an existing data link layer. However, certain types of connections exist that do not include a data link layer protocol over which IP can run. To enable TCP/IP to operate on these kinds of links, two special TCP/IP data link layer protocols have been created: the Serial Line Internet Protocol (SLIP) provides a layer two framing service for IP datagrams and the Point-to-Point Protocol (PPP) which defines a complete method for robust data link connectivity between units using serial lines or other physical layers. It includes numerous capabilities and features, including error detection, compression, authentication, encryption and much more.

2.1.1.2 The Internet layer

The Internet/network layer defines a protocol for data transmission through non-homogeneous networks. This protocol is called the Internet Protocol (IP) [3]; it enables communications across a vast and heterogeneous collection of networks that are based on different technologies, where any host computer that is connected to the Internet can communicate with any other computer that is connected to the Internet. The Internet therefore offers ubiquitous connectivity and the economies of scale that result from large deployment. IP was developed to provide for the connectionless transfer packets called (datagrams) which are received from the IP's upper-layer software to and from source and destination hosts and, to achieve this, it implements two functions: addressing and fragmentation.

Addressing, in the sense that in order for systems to locate each other in this distributed environment, nodes are given explicit addresses that uniquely identify the particular network the system is on, and uniquely identify the system to that particular network. When these two identifiers are combined, the result is a globally-unique address.

Every transmission medium has a limit on the maximum size of a frame it can transmit. As IP datagrams are encapsulated in frames, the size of IP datagram is also restricted. If the size of an IP datagram is greater than this limit, then it must be fragmented. Thus fragmentation is the breaking up of a single IP datagram into two or more IP datagrams of smaller size.

In IP, the component networks are interconnected by special packet switches called gateways or routers. Each router interface adapts to the particular attributes of the underlying network. IP routers direct the transfer of IP packets across an internet. After a routing decision is made, the packets are placed in a buffer to await transmission over the next network. In effect, packets from different users are statistically multiplexed in these buffers. The underlying networks are responsible for transferring the packets between routers. IP traditionally provides Best-effort service, that is IP makes every effort to deliver the packets but takes no actions when packets are lost, corrupted, delivered out of order, or even is undelivered, therefore, it does not guarantee to successfully send all the datagrams between the two hosts.

2.1.1.3 Transport layer

Two transport protocols will be mentioned here, TCP and UDP (User Datagram Protocol) build on best-effort service provided by IP to support a wide range of applications.

User Datagram protocol is a connectionless, unreliable transport service. It does not issue an acknowledgment to the sender upon the receipt of data. It does not provide order to the incoming packets, and may lose packets or duplicate them without issuing an error message to the sender. The only offering that UDP has is the assignment and management of port numbers to uniquely identify the individual applications that run on a network station and a checksum for error detection. UDP tends to run faster than TCP, for it has low overhead (8 bytes in its header compared to TCP's typical 40 bytes). Any application program that incorporates the use of UDP as its transport-level service must provide an acknowledgment and sequence system to ensure that packets arrive, and that they arrive in the same order as they were sent.

On the other hand, TCP [4, 5, 6] provides a logical full-duplex (two-way) connection between two application layer processes across the Internet on the top of the Internet

Protocol (IP). TCP provides these applications processes with connection-oriented, reliable, in-sequence, byte-stream service where TCP sends user data, or control data, in segments. A TCP segment as shown in Figure 2.2 will contain the TCP header and its data. TCP is responsible for breaking segments into IP packets before they are sent and for reassembling the packets when they arrive, furthermore it has to verify the correct delivery of data. TCP will be discussed in details later in section 2.3.

Bits **0**	**8**	**16**		**31**
Source Port			**Destination Port**	
Sequence Number				
Acknowledgment Number				
Data Offset	**Reserved**	**Code**	**Window**	
Checksum			**Urgent Pointer**	
Options				**Padding**
Data				

Figure 2.2 TCP Packet Header

2.1.1.4 Application layer

The top layer in TCP/IP is the *application layer.* This layer provides functions for users or their programs, and it is highly specific to the application being performed. It provides the services that user applications use to communicate over the network, and it is the layer in which user-access network processes reside. These processes include all of those that users interact with directly, as well as other processes of which the users are not aware of. This layer includes all applications protocols that use the host-to-host transport protocols to deliver data. Other functions that process user data, such as data encryption and decryption and compression and decompression, can also reside at the application layer.

A rich set of applications has been developed to operate on top of the TCP/IP. These applications include SMTP for e-mail service, FTP for file transfer, HTTP for web service, and RTP for real-time transfer of information such as voice and video and Boot Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) for the management of IP parameters on a network.

2.1.2 Client/Server Interaction

In a client/server environment, the client has the job of initiating contact, the server has to be ready (listening). With the server process running, the client process can initiate a TCP connection to the server. This is done in the client program by creating a socket. When the client creates its socket, it specifies the address of the server process, namely, the IP address of the server and the port number of the process. Upon creation of the socket, TCP in the client initiates a three-way handshake [3] which is completely transparent to the client and server programs, this will be explained later on. Socket facilities are provided to programmers through C system calls that are similar to function calls except that control is transferred to the operating system kernel once a call is entered.

A single, server-class computer can offer multiple services at the same time; a separate server program is needed for each service. Running many servers on a single computer is practical because a server does not consume computational resources while waiting for a request. In the real world, however, each simultaneous connection consumes some computational and memory resources of the server, even when idle, and this overhead grows with the number of clients.

Transport protocols assign each service a unique identifier. Both clients and servers specify the service identifier; protocol software uses the identifier to direct each incoming request to the correct server. A computer system that permits multiple application programs to execute at the same time is said to support concurrency which is fundamental to the client-server model of interaction because a concurrent server offers service to multiple clients at the same time, without requiring each client to wait for previous clients to finish. In a concurrent server, the main server thread creates a new service thread to handle each client. The server is constructed in two parts: one that accepts requests and creates a new thread for the request, and another that consists of the code to handle an individual request. When a concurrent server starts executing, only the first part runs. That is, the main server threads waits for a request to arrive. When a request arrives, the main thread creates a new service thread to handle the request. The service thread handles one request and then terminates. Meanwhile, the main thread keeps the server alive-after creating a thread to handle a request, the main thread waits for another request to arrive. If N clients are using a given service on a single computer, there are N+1 threads providing the service: the main thread is waiting for additional requests, and N service threads are each interacting with a single client.

TCP requires each client to choose a local protocol port number that is not assigned to any service. When it sends a TCP segment, a client must place its local protocol port number in the SOURCE PORT field and the protocol port number of the server in the DESTINATION PORT field. On the server's computer, TCP uses the combination of source and destination protocol port numbers (as well as client and server IP addresses) to identify a particular communication. Thus, messages can arrive from two or more clients for the same service without causing a problem. TCP passes each incoming segment to the copy of the server that has agreed to handle the client.

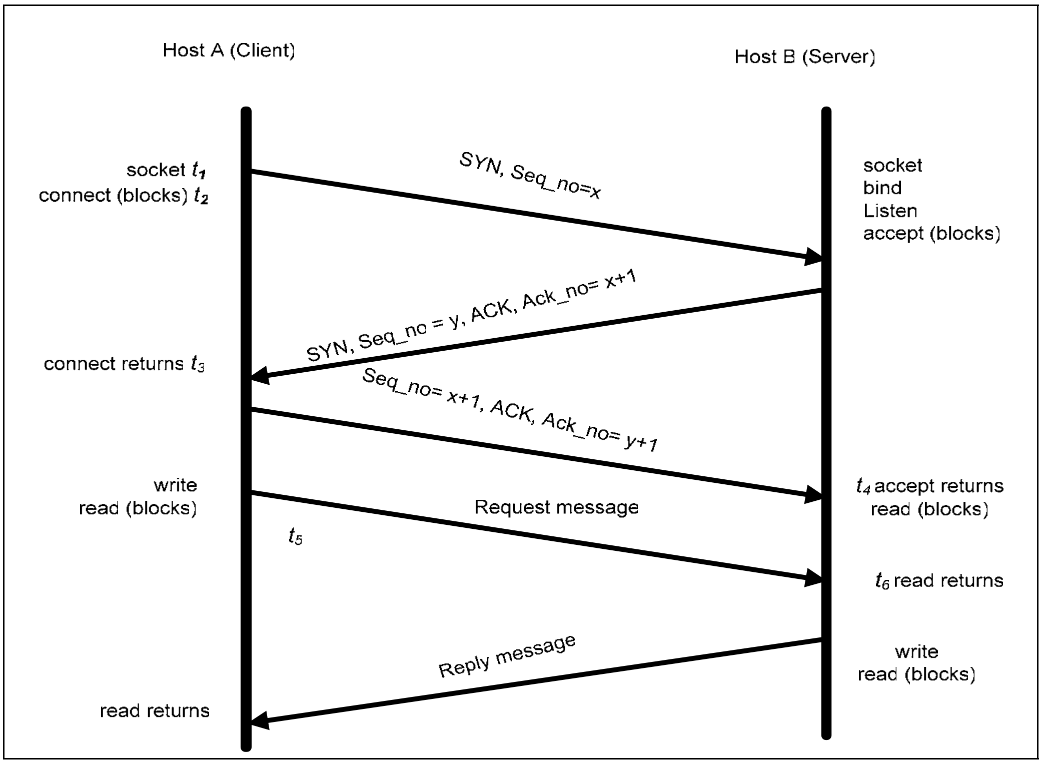


Figure 2.3 Client/server application process actions and TCP [3]

In three-way Handshake [3] shown in Figure 2.3, The server must first carry out a passive open to indicate to TCP that it is willing to accept connections using the socket system calls socket, bind, listen and accept therefore it will listen for connections on a certain port number that the client knows. The client initiates the session by performing an active open, making a socket call (t1) that creates a socket on the client side and then a connect call (t2) that initiates the TCP connection three-way handshake. When the server's TCP receives the first SYN, it returns a segment with an ACK and its own SYN. When the client's TCP receives this segment, connect returns (t3) and the client's TCP sends an ACK. Upon receiving this ACK, accept returns (t4) in the server, and the server is ready to read data. The client then issues a write call (t5) to send a request message. Upon receipt of this segment by the TCP module in the server, read returns (t6), and the request message is passed to the server. Subsequently, the server sends a reply message.

A client establishes a connection on a socket by calling connect, clients and servers may transmit data using write usually for connection-oriented mode or send to usually for the connectionless mode, then for closing a connection if the socket is no longer in use, the application can call close to terminate a connection and return system resources to the operating system. The functions of sockets are shown in Figure 2.4.

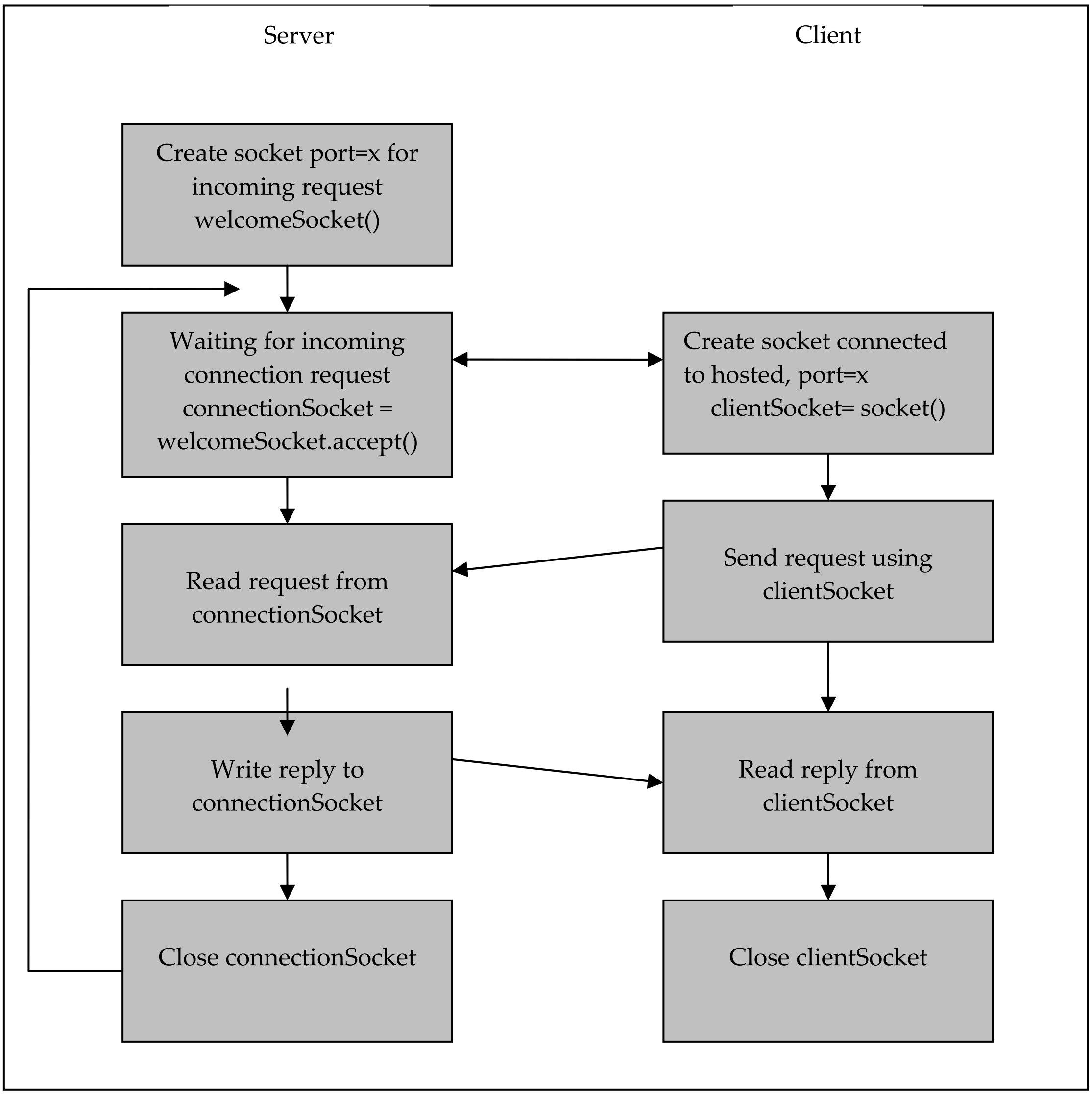

Server
Client
Create socket port=x for incoming request welcomeSocket()
Waiting for incoming connection request connectionSocket = welcomeSocket.accept()
Create socket connected to hosted, port=x clientSocket= socket()
Read request from connectionSocket
Send request using clientSocket
Write reply to connectionSocket
Read reply from clientSocket
Close connectionSocket
Close clientSocket

Figure 2.4 Client/server application, using connection-oriented transport services [7]

2.2 Mobile Computing

Mobile networking [3] is becoming increasingly important as portable devices such as personal digital assistance (PDAs) and Notebook computers are becoming more powerful and less expensive coupled with people's need to be connected whenever and wherever they are. The link between the portable device and the fixed communication network can be wireless or wired. If a wireless link is used, the device can utilise a radio or infrared channel. Radio channels can traverse longer distances without the line-of-sight requirement, but introduce electromagnetic interference and are often subject to federal regulations (e.g. federal communication commission (FCC)). Infrared channels are often used in shorter distances. A wireless connection enables a user to maintain its communication session as it roams from one area to another, providing a very powerful communication paradigm.

Mobile computing is also a technological term used to describe the blend of technologies to provide access for computer users to their computing environments either by the availability of fixed mobile devices whenever the user roams or travels, or by carrying mobile devices with the user when he/she travels [8]. To provide mobility a computing device (laptop or PDA) can be used that may have discontinuous network connectivity, or another form of mobility is introduced where the user's applications are mobile, therefore no additional hardware/computing platform is required and the user is able to bring up his/her applications on any nearby machine exactly as they appeared when they last brought up this way; this form is called "teleporting" [9].

Ubiquitous computing was first envisioned by Mark Weiser who envisioned a world full of connected computers and the user did not need to carry any form of hardware because of the fact that information is accessible everywhere. Ubiquitous computing envisions computation primarily in the background where it may not even be noticed [10].

One of the areas of research in ubiquitous computing is "Mobile Applications" to achieve ubiquitous computing where the research is done to enable applications to move with the user. Therefore, mobility can be seen as the migration of applications from a platform or the migration of the platform with the user to another site where the platform re-establishes its network connections to a new server in the new local site.

Looking back in time as described in [11] the computational model 50 years ago "was where programs are executed on a single fixed piece of hardware and…information tends to be inaccessible when users change location". The introduction of Java and mobile codes in addition to the popularity and the growing speed of the Internet with its ubiquitous nature, have caused a radical shift to a new computational model where the processes (applications) are free to migrate with the users.

The notion of the mobile application introduced in [12] is one where user interface can be dynamically mapped onto the resources of the surrounding computer and communications facilities. The demand of viewing your personal computing environment had pushed the researches in this area where we are living in the age of the Internet and virtual shared workspaces.

Teleporting and its free source successor VNC described later in chapter three are examples of global smart personalisation. These systems allow a user to preserve a GUI session as the user moves from one thin client to another. As [13] explained it, the system implements personalisation in the sense that any computer becomes your computer just by connecting to the networked server. In teleporting and Xvnc the user can login and call his private session, even if someone is using the workstation without disturbing the existing session.

As mentioned above, there are two different kinds of mobility: user mobility and device mobility. User mobility refers to a user who has access to the same or similar telecommunication services at different places, i.e., the user can be mobile, and the services will follow him. Examples for mechanisms supporting user mobility are simple call-forwarding solutions known from the telephone or computer desktops supporting roaming (i.e., the desktop looks the same no matter which computer a user uses to log into the network).

With device mobility, the communication device moves (with or without a user). Many mechanisms in the network and inside the device have to make sure that communication is still possible while the device is moving. A typical example for systems supporting device portability is the mobile phone system, where the system itself hands the device from one radio transmitter (also called a base station) to the next if the signal becomes too weak. A communication device can exhibit one of the following characteristics:

1. Fixed and wired: this configuration describes the typical desktop computer in an office. Neither weight nor power consumption of the devices allow for mobile usage. The devices use fixed networks for performance reasons.
2. Mobile and wired: many of today's laptops fall into this category; users carry the laptop from one hotel to the next, reconnecting to the company's network via the telephone network and modem.
3. Fixed and wireless: this mode is used for installing networks, e.g., in historical buildings to avoid damage by installing wires, or at trade shows to ensure fast network setup. Another example is bridging the last mile to a customer by a new operator that has no wired infrastructure and does not want to lease lines from a competitor.
4. Mobile and wireless: this is the most interesting case. No cable restricts the user, who can roam between different wireless networks.

Wireless technology can eliminate many of the wires and, in the process, simplify the installation and movement of equipment, as well as provide connection between computers. Many applications can benefit from wireless networks and mobile communications: vehicles, emergencies, business, and replacement of wired networks, infotainment, location dependent services and mobile and wireless devices.

With all advantages, mobile computing introduces an environment quite different from the one found in fixed networks due to scarce radio bandwidth and intermittent connectivity.

Wireless networks operate on the same hierarchy as their wired counterparts; small networks of three or more devices are referred to as Wireless LANs (WLANs), while the global wireless network is referred to as the wireless Internet. Other basic types of wireless networks include the Wireless Personal Area Network (WPAN), the Wireless Metropolitan Area Network (WMAN), and the Wireless Wide Area Network (WWAN).

WLANs use electromagnetic waves (typically radio or infrared), to enable communication between devices in a limited area. Spread spectrum technology, based on radio transmission is most commonly used to deploy WLANs today.

In wired LAN the MAC address specifies the physical location of a station, since users are stationary. In wireless LAN, the MAC address identifies the station but not the location, since the standard assumes that stations can be portable or mobile. A station is portable if it can move from one location to another but remains fixed while in use, a mobile station moves while in use. IEEE 802.11, the Wi-Fi standard, denotes a set of Wireless LAN/WLAN standards developed by the IEEE for wireless LAN technology; An 802.11 LAN is based on a cellular architecture where the system is subdivided into cells, where each cell (called Basic Service Set or (BSS)) is controlled by a base station (called Access Point or (AP)).

Wireless WANs, which can bridge branch offices of a company, cover a much more extensive area than wireless LANs and it generally use digital cellular phone networks to enable notebooks and handheld computers to access the Internet across extensive geographic areas.

Unlike WLANs, which offer limited user mobility and instead are generally used to enable the mobility of the entire network, WWANs facilitate connectivity for mobile users such as the travelling businessman. Unlike WLANs, which are unlicensed and typically administered privately by the customer, WWANs are generally operated by public carriers, and use open standards such as AMPS, GSM, TDMA, and CDMA. In general, WWANs allow users to maintain access to work-related applications and information while away from their office.

In wireless WANs, communication occurs predominantly through the use of radio signals over analogue, digital cellular, or PCS networks, although signal transmission through microwaves and other electromagnetic waves is also possible. Today, most wireless data communication takes place across 2G cellular systems such as TDMA, CDMA, and GSM, or through packet-data technology over old analogue systems such as CDPD overlay on AMPS. Data services provided by W-WANs allow for a rather low link speed; error losses, changing line rate and variable delays present additional challenges for an efficient data transport.

2.3 Mobile IP [14]

The number of mobile hosts is increasing with the emergence of lightweight digital personal assistants and laptop computers capable of wireless communication.

Routing mechanisms rely on the assumption that each network node will always have the same point of attachment to the Internet, and that each node’s IP address identifies the network link where it is connected. Core Internet routers look at the IP address prefix, which identifies a device’s network. At the network level, routers look at the next few bits to identify the appropriate subnet. At the subnet level, routers look at the bits identifying a particular device. Therefore, if you disconnect a mobile device from the Internet and want to reconnect through a different subnet, you have to configure the device with a new IP address, and the appropriate netmask and the default router. Otherwise, routing protocols have no means of delivering packets because the device’s

IP address does not contain the necessary information about the current point of attachment to the Internet.

With true IP level mobility, nomadic Internet users would be able to access network resources and exchange data with each other in a similar manner. There is a severe problem with higher layer protocols like TCP which rely on IP addresses, because changing the IP address while still having a TCP connection open means breaking the connection. A TCP connection cannot survive any address change because it is based on the following tuple to identify its connection (source IP address, source port, destination IP address, destination port), also known as a socket pair.

Mobile IP is the most prominent way to introduce mobility to the Internet protocol based communications. The idea is to make communicating hosts unaware of the underlying movement between the subnets. This allows applications to operate as if the hosts were stationary.

2.3.1 Architecture

Mobile IP performs mobility management by introducing two network entities: home agent (HA) and foreign agent (FA) as shown in Figure 2.5.

Mobile IP allows portable devices called mobile Nodes (MNs) to roam from one area to another while maintaining the communication sessions. One requirement in Mobile IP is that a legacy host communicating with an MN and the intermediate routers should not be modified. This requirement implies that an MN must continuously use its permanent IP address even as it roams to another area. Otherwise, existing sessions will stop working and new sessions should be restarted when an MN moves to another area and the higher layer protocols lose the end-to-end connection they need for exchanging packets. A permanent IP address called a home address is allocated from the home network of the mobile node. Packets destined for the mobile node are routed with the home address to the home network. The packets are intercepted by a ***home***

agent, a network entity residing at the home network. The home agent tunnels packets to the current location of the mobile node. Whenever the mobile node moves to a new subnet, it registers its new location to the home agent.

When an MN moves to a foreign network, the MN obtains a care-of address from the ***foreign agent*** (FA) and registers the new address with it's HA. The care-of address reflects the MN's current location and is typically the address of the FA. Once the HA knows the care-of address of MN, the HA can forward the registration packet to the MN via the FA. A Correspondent Node (CN) is the host with which the Mobile Node is trying to communicate, on the Internet.

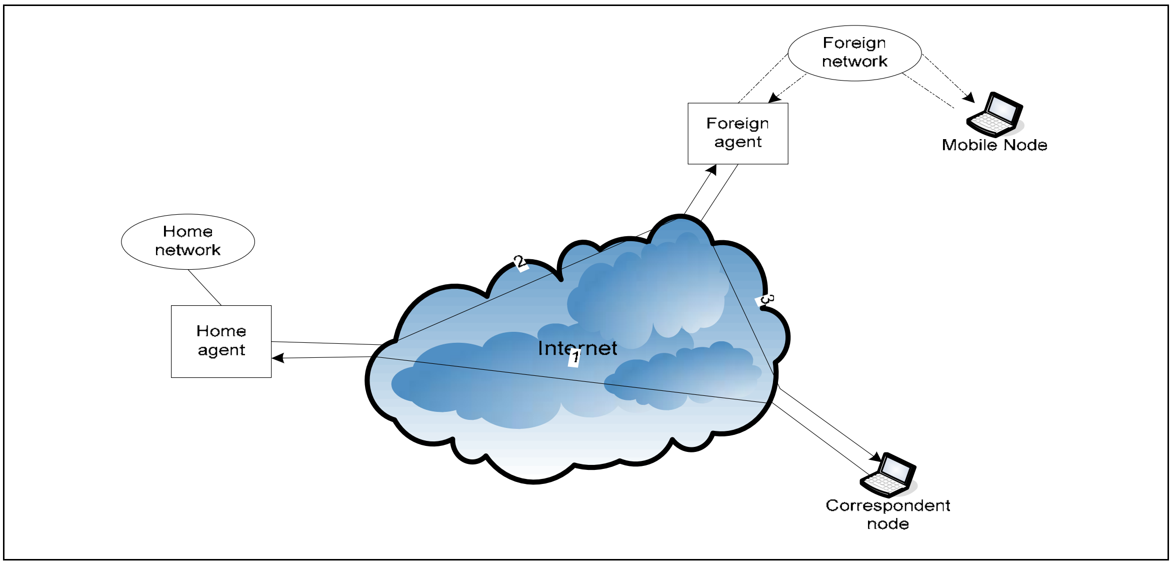


Figure 2.5 Routing for mobile hosts

Unfortunately, the HA cannot directly send packets to the MN in a foreign network in a conventional way (i.e., by using the care-of address as the destination address of the IP address, the packet final destination will be the FA rather than the MN). The problem is solved by providing a tunnel between the HA and the FA and it is implemented by encapsulating each IP packet at the HA with an outer IP header containing the HA's address as the source IP and the care-of address as the destination IP address. When the FA receives the packets, the FA decapsulates the packet that produces the original IP packet with the original IP address. The FA can then deliver the packet to the MN.

Packets transmitted by the MN to the Correspondent Node (CN), use a normal IP packet format with the MN’s address as the source IP address and the CN’s address as the destination IP address. These packets follow the default route. The route travelled by the packet from the CN to the MN is typically longer than that from the MN to the CN. One solution is when the HA receives a packet from a CN destined to an MN, it tunnels the packet to the current care-of address. HA forwards these new IP packets to MN’s CoA, then FA or MN itself recovers the original IP packets. It also sends a **binding message** back to the CN containing the current care-of address. The CN can save this message in its **binding cache** so that future packets to the MN can be directly tunnelled to the care-of address.

When a mobile node moves and registers with a new foreign agent, the base Mobile IP protocol does not notify the mobile node's previous foreign agent. IP datagrams intercepted by the home agent after the new registration are tunnelled to the mobile node's new care-of address, but datagrams in flight that had already been intercepted by the home agent and tunnelled to the old care-of address when the mobile node moved are likely to be lost and are assumed to be retransmitted by higher-level protocols if needed. The old foreign agent eventually deletes its visitor list entry for the mobile node after the expiration of the registration lifetime.

Route Optimization provides a means for the mobile node's previous foreign agent to be reliably notified of the mobile node's new mobility binding, allowing datagrams in flight to the mobile node's previous foreign agent to be forwarded to its new care-of address. This notification also allows any datagrams tunnelled to the mobile node's previous foreign agent, from correspondent nodes with out-of-date binding cache entries for the mobile node, to be forwarded to its new care-of address. Finally, this notification allows any resources consumed by the mobile node at the previous foreign agent (such as radio channel reservations) to be released immediately, rather than waiting for its registration lifetime to expire.

In Mobile IP, the HA and the FA periodically send agent advertisement message on home network and foreign network respectively to notify mobile host (MN) their existence. The agents also send agent advertisement message in response to MN's active solicitation message. MN judges whether it is on the home network or the foreign network according to the agent advertisement message. When MN is on its home network, it communicated with others using the normal IP; otherwise, it needs to get an IP address care-of address (CoA), then sends registration message to HA to report its current location.

2.3.2 Handover

The increasing number of Internet users in combination with the evolution of IP-based applications has created a strong demand for wide-area broadband access to IP services. A future wireless Internet is expected to consist of different types of wireless networks, each providing varying access bandwidth and coverage level. Today, the natural trend is to utilize high-bandwidth wireless local area networks (WLANs) such as IEEE 802.11 in hotspots and switch to wireless wide area networks (WWANs) such as General Packet Radio Service/Universal Mobile Telecommunications System (GPRS/UMTS) networks when the coverage of WLAN is not available or the network condition in WLAN is not good enough. We refer to such a procedure as *vertical handover*. By combining the wide coverage of next-generation cellular systems with the advantage of high bandwidth in WLANs, users can make the most of wireless IP communication. Whereas, handoff schemes that only deal with the switch between base stations (BSs) or access points (APs) in the homogeneous wireless system are usually called *horizontal handover.*

It is worth to mention that this thesis will concentrate on horizontal handover in WWANs.

Moving from one subnet to another causes a handover also called handoff in North America. During a handover, a mobile node's point of attachment to the fixed network is transferred to another. This causes a disruption in the ongoing data flow, in the transport layer, TCP protocol interprets the break caused by the handover as network congestion. The congestion control mechanisms delays packet transmission even more.
Four stages in the operation of Mobile IP:
1- The Mobile Node has to setup a link layer connection to a new point of attachment. (Layer 2 handover)
2- Then the movement has to be detected in the network layer. This can be done by employing network layer mechanisms or by sharing information between the link and network layer.
3- After that, a temporary IP address called a care-of-address must be acquired from the new subnet.
4- The MN sends a registration request to the HA to update the mobility binding with the newly acquired CoA in order for packets to be redirected to the mobile node.
5-when the HA received the registration request, it answers with a registration reply and starts to tunnel arriving packets to the new care-of address of the MN.

If handover is not handled properly, it causes perceivable degradation of quality of service. Especially real-time and delay-sensitive applications suffer from the interrupted data flow. The handover algorithm seeks to execute handovers as efficiently as possible to minimise packet drop and latency.
The efficient execution of the handover requires two things:
1- The time that the MN is unable to exchange packets with the corresponding nodes should be minimal.
2- The number of packets dropped because of the handover should be minimal.

Two important factors affect the handover process: the handover initiation algorithms and handover decisions protocols.

Firstly, there are four main handoff initiation algorithms: Relative signal strength, relative signal strength with threshold, relative signal strength with hysteresis, and relative signal strength with hysteresis and threshold. For the purpose of this thesis we will briefly explain one of them, however for more information on all of them refer to [15].

Relative Signal Strength with Hysteresis and Threshold (RSSHT) handover algorithm describes the conditions to start handoff which is also shown in Figure 2.6:

As the MN moves from old AP to new AP, the signal-to-noise Ratio (SNR) from old AP decreases while SNR from new AP increases.

IF SNR(d) ≥ Tcs (cell search threshold) **THEN**

a handoff is not required (normal state) when

IF SNRold(d) < Tcs (Position 1) **THEN**

MN enters the discovery state and initiates an active discovery.

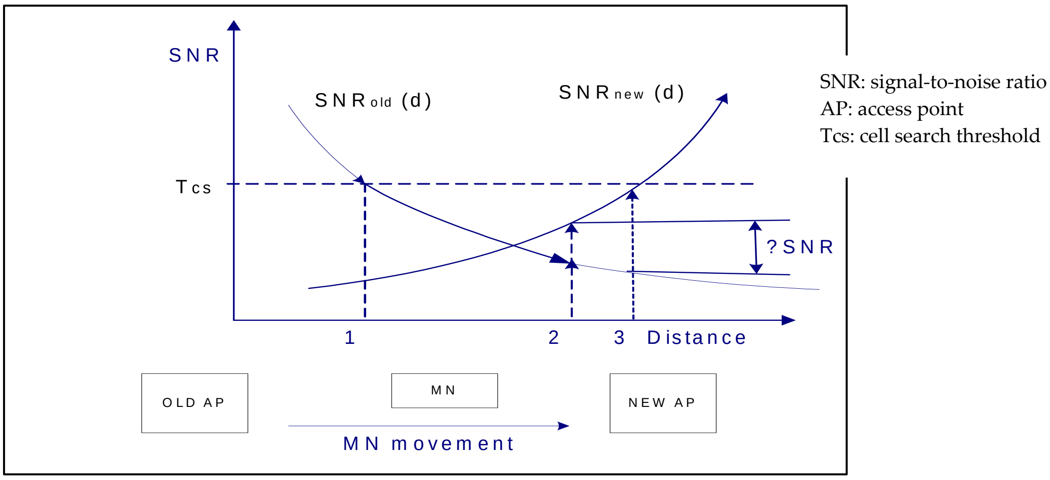
While the SNR value keeps lower than Tcs , the MN will achieve a short-scan active discovery for every 2 seconds (ORiNoCo systems)

After scan latency, a comparison is made by the MN at some point during the trip to new AP.

IF | SNRnew(d) –SNRold (d)| > Δ SNR (hysteresis threshold) **THEN**

the MN will definitely change its association from old t new AP (position 2) .

After re-association, the MN may keep standing in the discovery phase or it may pass directly to a normal operation state (position 3), depending on the values for Tcs and ΔSNR avoids ping-pong effect in handovers.



SNR: signal-to-noise ratio
AP: access point
Tcs: cell search threshold

Figure 2.6 Decision points during layer 2 handoff process

Secondly, the handoff decision protocols used in various cellular systems are either network-controlled handover (NCHO), mobile-assisted handover (MAHO) or mobile-controlled handover (MCHO).

NCHO is used in first generation cellular systems such as; Advanced Mobile Phone System (AMPS) where the mobile telephone switching office (MTSO) is responsible for the overall handoff decision [16]. In NCHO, the network handles the necessary RSS measurements and handoff decision. The handoff execution time is on the order of many seconds because of the high network load [17].

In NCHO, the load of the network is high since the network handles all of the processes itself. In order to reduce the load of the network, the MN is responsible for making RSS measurements and sending them periodically to BS in MAHO. Based on the received measurements, the BS or the mobile switching centre (MSC) decides when to handoff [15, 18]. MAHO is used in the Global System for Mobile Communications (GSM). The handoff execution time is about 1 sec [17, 18].

MCHO extends the role of the MS by giving overall control to it. Both, MN and BS, make the necessary measurements, and the BS sends them to the MN [15]. Then, the MN decides when to handoff based on the information gained from the BS and itself.

Digital European Cordless Telephone (DECT) is a sample cellular system using MCHO with 100-500 ms handoff execution time [17, 18].

2.3.3 Cross-layer optimisation

Mobile IP has been designed without any assumption concerning the underlying link- layer. This separation between layers results in the following sources of handoff delays [19]:

1. MN involved in a handoff may only begin the registration process after the link-layer (L2) handoff to the new foreign agent has been completed.
2. As the messages generated by the registration process need some time to propagate through the network, the MN is unable to send or receive packets at that time. This may lead to a handover latency that is unacceptable for the support of real-time services. Therefore IETF proposed the low latency handoff schemes based on information of the L2 handoff process received using L2 triggers.

Designing without any assumption regarding underlying layers, allows the widest possible applicability and a clean separation between Layer 2 and Layer 3 of the protocol stack. However, this layer separation results in lower performance. Indeed, the MN may only communicate with a directly connected FA and therefore it can only start the registration process after completion of the L2 handoff. Moreover, the MN is unreachable during the registration process, a property that may contribute to a non-negligible handoff latency and packet loss.

In [20], it is proposed using the concept of cross-layer optimization [21] it adds the function into Mobile IP module and make no modification to the higher layer protocol. When MN serves as the sender, a buffer is deployed at IP layer to cache the segments passed down by TCP layer. By monitoring the returned ACKs, it can deduce which segments have been received by the receiver and delete those. After handover, MN retransmits the segments remained in the buffer.

When MN servers as the receiver, the mobile IP layer saves one copy of the latest Selective Acknowledgment (SACK) sent by TCP layer. After handover, MN sends this same SACK thrice. Thus the sender can utilise this information to fast retransmit the lost ones to avoid timeout retransmission.

Handoff latency results in packet losses and severe end-to-end TCP performance degradation as TCP, perceiving these losses as congestion, causes source throttling or retransmission.

In WLAN, Handover management is divided between link layer (layer 2) handover algorithms wireless network adaptors and a network layer (layer 3) handover algorithm. The link layer algorithm decides the access point to associate with and the timing of the L2 handover. The network algorithm chooses the interface to use, acquire a CoA from the visited subnet, and handles the mobile IP registration process. The network layer may or may not be aware of the link layer handover. If not, the network layer handover algorithm has to detect the movement by other means.
It is worth mentioning that in [20] the interruption time due to handover can reach 6 seconds, even grievous 12 seconds, during which, there may be several timeout retransmissions. How to eliminate TCP timeout retransmission caused by these lost packets is a crucial issue.

One method of a handover is a hard handover where mobile node cannot sustain simultaneous communication with the new and old access point (AP). This implicates that the MN will suffer from a temporal disconnection that will impede the exchange of data packets with other stations during the handoff. Such disruption may be present during all the handoff or just parts of it.

2.4 TCP performance

2.4.1 Traditional TCP [7, 22, 23]

TCP sends user data or control data in segments not exceeding the Maximum Segment Size (MSS) of the connection. Segments are assigned a unique sequence number, to ensure the delivery of data in the proper order. The receiver sends an acknowledgment (ACK) upon reception of a segment. Therefore, through the use of both acknowledgements and sequence number, TCP can detect errors or lost data and can trigger retransmission until the data is received, complete and without errors.

Arrived segments that do not begin at the number of the next unacknowledged segment are called out-of-order data. As a response to out-of-order segments, TCP sends duplicate acknowledgments (DUPACK) that carry the same acknowledgment number as the previous ACK. In combination with a retransmission timeout (RTO) on the sender side, ACKs provide reliable data delivery [5]. The retransmission timer is set up based on the smoothed round trip time (RTT) and its variation. RTO is backed off exponentially at each unsuccessful retransmit of the segment [24]. When RTO expires, data transmission is controlled by the slow start algorithm described below.

One function provided by TCP in order to manage a connection is ***flow control***. Flow control is used in order to avoid the sender to fill the buffer of the receiver, in practice it has been designed to control the speed of the sender basing on the speed used by the receiver to read the incoming data and allows a TCP receiver to control the rate at which the sender transmits information so that the receiver buffers do not overflow. Flow control is achieved using the sender's window, called *receive window*, that is periodically set to the value of the receiver’s buffer. TCP implements the flow control based on a sliding window [16]. When the total size of outstanding segments, segments in flight (FlightSize), exceeds the window advertised by the receiver, further transmission of new segments is blocked until ACK that opens the window arrives.

Another feature of TCP is to have each sender limit the rate at which it sends traffic into its connection as a function of perceived network congestion. If a TCP sender perceives that there is little congestion on the path, then the sender reduces its send rate. This is called ***congestion control*** algorithm [22], which uses the *congestion window* (cwnd) to control the rate at which packets are sent and as a current estimation of the available capacity in the network. A TCP connection starts with a slow-start phase by sending out the initial window where the standard currently allows the initial window of one or two segments [6]. At any point of time, the sender is allowed to have no more segments outstanding than the minimum of the advertised and congestion windows. The sender sends one packet and waits for acknowledgement. Upon reception of an acknowledgment, the congestion window is increased by one (congestion window= 2), thus the sender is allowed to transmit the number of acknowledged segments plus one. After arrival of the two corresponding acknowledgments, the sender again adds 2 to the congestion window equals 4, this roughly doubles the congestion window per RTT, and this is called the exponential growth of the congestion window in the ***slow start*** mechanism. The slow start ends when a segment loss is detected or when the congestion window reaches the slow-start threshold also known as congestion threshold (ssthresh). When the slow start threshold is exceeded, the sender is in the congestion avoidance phase and increases lineary the congestion window roughly by one segment per RTT. As soon as the congestion window reaches the congestion threshold, further increase of the transmission rate is only linear by adding 1 to the congestion window each time the acknowledgments come back.

Linear increase continues until a time-out at the sender occurs due to a missing acknowledgment, or until the sender detects a gap in transmitted data because of continued acknowledgments for the same packet. In either case the sender sets the congestion threshold to half of the current congestion window. The congestion window itself is set to one segment and the sender starts sending a single segment. The exponential growth starts once more up to the new congestion threshold, then the window grows in linear fashion.

TCP recovery was enhanced by the fast retransmit and fast recovery algorithms to avoid waiting for a retransmit timeout every time a segment is lost [25]. Recall that DUPACKs are sent as a response to out-of-order segments. Because the network may re-order or duplicate packets, reception of a single DUPACK is not sufficient to conclude a segment loss. A threshold of three DUPACKs was chosen as a compromise between the danger of spurious loss detection and a timely loss recovery. Upon the reception of three DUPACKs, the fast retransmit algorithm is triggered. The DUPACKed segment is considered lost and is retransmitted. At the same time congestion control measures are taken; the congestion window is halved. The fast recovery algorithm controls the transmission of new data until a non-duplicate ACK is received. The fast recovery algorithm treats each additional arriving DUPACK as an indication that a segment has left the network. This allows inflating the congestion window temporarily by one MSS per each DUPACK. When the congestion window is inflated enough, each arriving DUPACK triggers a transmission of a new segment, thus the ACK clock is preserved. When a non-duplicate ACK arrives, the fast recovery is completed and the congestion window is deflated.

2.4.2 TCP in Wireless environments

These days more and more people enjoy the advantages of the portability and flexibility of carrying their workstations in the form of laptops, notebooks, and PDA handsets. To meet the needs of this new set of users, existing computing environments based on fixed networks are being extended into the mobile domain to meet users demand the mobility of hosts where they expect that the hosts can change their locations continuously without interrupting current communication sessions. 3G and wireless networks are seen as a major revenue stream by providing an additional level of flexibility and service to the user [26].

TCP is the defacto standard of reliable transport protocol, one implicit assumption of TCP is that the packet losses are caused by network congestion. Each time packet loss occur TCP will invoke congestion control mechanisms to decrease the TCP sending rate. This assumption is valid in wired networks and it does work well, so TCP is widely employed, however, this assumption is not always right especially when the packets losses are caused by user mobility such as handover etc. The packet losses during handover will make the interruption time longer and deteriorate TCP performance.

Compared to WLANs, W-WANs such as Cellular Communications (CC) systems exhibit higher transmission and propagation delays due to the lower bit rates and longer distances involved. The outdoor CC environment is also harsher, with multipath fading caused by buildings and hills, leading to high error rates. Due to the real time requirements of voice telephony, *Forward Error Correction* (FEC) information is added to each frame, allowing damaged frames to be recovered without retransmissions. Bit errors due to fading are usually bursty, therefore bits from multiple consecutive frames are interleaved before transmission, so as to evenly spread the error bursts and increase the probability of successful recovery.

Digital CC systems are interconnected to other networks using an *Interworking Function* (IWF) [27], located at the boundary of the CC system. To interface with analogue telephony networks, the IWF converts analogue waveforms to digital data and vice versa.

To interface with ISDN, the IWF performs rate adaptations and frame conversions, since even though both networks are digital, their implementations differ. In order to directly interoperate with packet data networks such as the Internet, the IWF may also serve as a gateway, as shown in Figure 2.7. In this configuration, the wireless host and the IWF communicate via a base station, which simply relays frames between the two. A *Radio Link Protocol* (RLP) is used between the wireless host and the IWF, offering IP datagram segmentation and reassembly [28]. As a result, the wireless host may exchange IP datagrams with any host on the Internet, using the IWF for routing purposes. The RLP may also provide error recovery in order to hide wireless losses from the Internet [29].

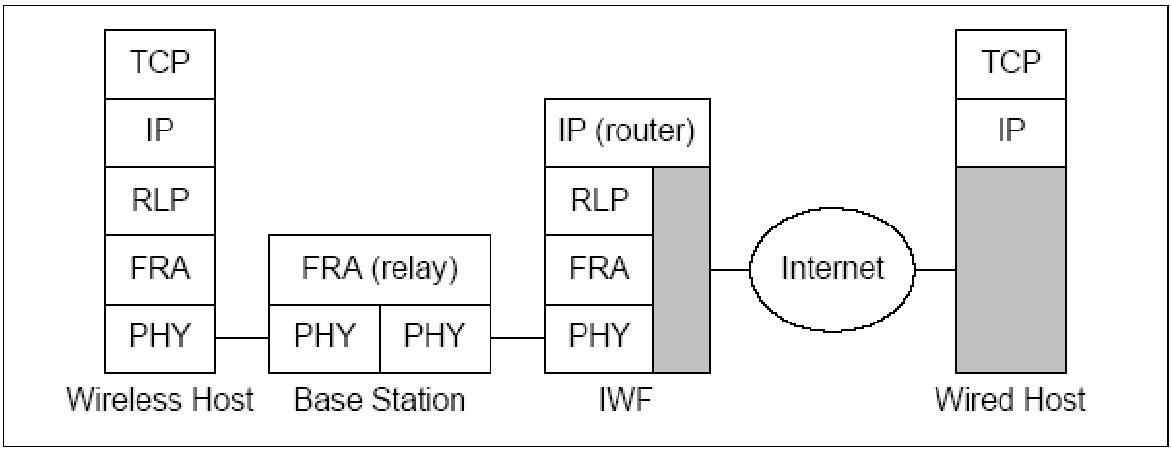


Figure 2.7 Connectivity between CC systems and the Internet [30]

There are a number of possible reasons for such type of delays in W-WANs [31]:

One of the reasons of delay is ***Wireless Link delay***. This is the most widely known source of varying delays as it presents a possibility for competing of link-level and end-to-end protocols in recovering error losses on a data link [32]. For example, an error burst requiring a large number of link-level retransmissions may be caused by a partial loss of the radio signal while driving into a tunnel. If the persistence of link-level error recovery exceeds the typical RTO of TCP over the given connection, spurious timeouts may result.

Although studies report that cases of competing error recovery are infrequent in the basic GSM data service [33], the situation may be different for other W-WANs. Some wireless networks can include two or more layers of protocols capable of error recovery at the link level. For example, the GPRS wireless network includes both the Radio Link Control (RLC) protocol operating with small-sized frames at the lower level and the Logical Link Control (LLC) protocol operating with IP datagrams at the higher level [34]. Both protocols can be used in reliable or unreliable mode and the maximum number of retransmissions can be set as a network parameter. Optimally configuring the persistence of individual protocol sublayers may be a difficult task for a network operator. Thus, the TCP protocol in not guaranteed against operation over a highly persistent link introducing long delays in data transfer.

A second reason of delay is ***Handovers.*** TCP session when a mobile user in a wireless cellular network moves from one cell to another, all necessary information must be transferred from the previous base station (BS) to the new base station which might cause a short duration of disconnection (typically of the order of several hundreds of milliseconds) during which no transmission takes place. During a handover the mobile terminal may have to perform some time-consuming actions before data can be transmitted in a new cell. These include, for example, collection of signal quality, transmitting it to the new base station, authentication, etc.

Thus, as the explosive growth of the mobile Internet, especially in urban areas, cell sizes may become small because of higher frequency by adopting high bandwidth and high density of mobile users, small cell size leads to frequent user handover, though BER might be reduced. Therefore, the performance degradation due to handover tends to be more serious to mobile users moving frequently than due to high BER. This “handoff" scenario triggers TCP congestion control mechanisms which results in reduced end-to-end throughput.

Many W-WANs in such a case try to provide seamless mobility, that is internally re-route packets from the old to the new base station at the expense of additional delay. As the result, the data transfer can be suspended for tens of seconds.

Mobility itself can cause packet loss. There are many situations where a soft handover from one access point to another is not possible for a mobile end-system. For example, when using mobile IP, there could still be some packets in transit to the old foreign agent while the mobile node moves to the new foreign agent. The old foreign agent may not be able to forward those packets to the new foreign agent or even buffer the packets if disconnection of the mobile node takes too long. This packet loss has nothing to do with wireless access but is caused by the problems of rerouting traffic.

Another reason of delay is ***Link Blackout and Blocking by high-priority traffic***. Wireless links are prone to temporal interruptions of service. A typical example is loss of radio coverage; this might happen due to driving in a tunnel or moving away from the serving access point. Blackouts cause a situation when, for a period of time, no user data is successfully transmitted through the link.

When packet-switched and voice calls have to co-exist in a W-WAN network, in most cases the network operator assigns a higher priority to voice calls. An incoming voice call can temporarily pre-empt radio resources from packet-switched traffic, thus causing a delay in data transfer in the order of tens of seconds. Currently the support for Quality of Service (QoS) is being introduced into packet switched W-WANs. Interactive traffic, for example web browsing, may have higher priority over best-effort bulk data transfers. There can be situations when the lower-priority traffic is delayed when higher-priority connections become active.

Also ***Transmission losses on wireless link***, the TCP reaction in case of packet losses due to random errors is that sender can perform a retransmission of what appears to be missing segments using the "fast retransmit" algorithm without waiting for the retransmission timer to expire after receiving 3 duplicate ACKs. On the other hand, in case of packet losses due to handover, the sender may have to wait for a retransmission timer to expire and perform "slow start" since in-flight packets are forwarded to the previous access point to which the mobile node attached before movement and result in packet losses.

While slow start is one of the most useful mechanisms in fixed networks, it drastically decreases the efficiency of TCP if used together with mobile receivers or senders. The reason for this is the use of slow start under the wrong assumptions. From a missing acknowledgment, TCP concludes a congestion situation. While this may also happen in networks with mobile and wireless end-systems, it is not the main reason for packet loss.

TCP does not react well to large delays (several times the usual RTT) that occur suddenly. Without a chance to adapt its retransmission timer to such a delay, TCP has to assume that outstanding segments were lost and retransmits them.

In [35] the effects of large sudden delays on the TCP performance are studied using tcpdump [36] program. When a sudden delay that exceeds the current value of TCP retransmission timer occurs in the data transfer, TCP times out and retransmits the oldest outstanding segment. Since data segments are delayed but not lost, the retransmission is unnecessary and the timeout is spurious.

This happens due to the retransmission ambiguity problem as the ACK bears no information which segment, original or retransmitted, has generated it. Encouraged by arriving ACKs, TCP retransmits all outstanding segments using the slow start algorithm. Also, a number of new segments allowed by the congestion window are transmitted. Such retransmission policy is refereed to as go-back-N since the sender forgets about all segments it has earlier transmitted.

2.5 TCP Improvements

TCP cannot be directly applied in wireless networks in which packet loss may be induced by higher Bit Error Rate (BER) or handover than congestion. It assumes that such packet loss is caused by network congestion and initiates congestion control procedures. (E.g. reduction of its congestion window (cwnd)), this incorrect assumption causes TCP to perform poorly in wireless environments.

Several schemes have been proposed for improving TCP performance over wireless links, an overall view of the different proposed mechanisms to improve TCP performance in wireless networks is shown in Figure 2.8. In the next section we will briefly mention four proposed solutions, one related to link-level approaches, while the other three are related to transport-level approaches which we are mainly interested in.

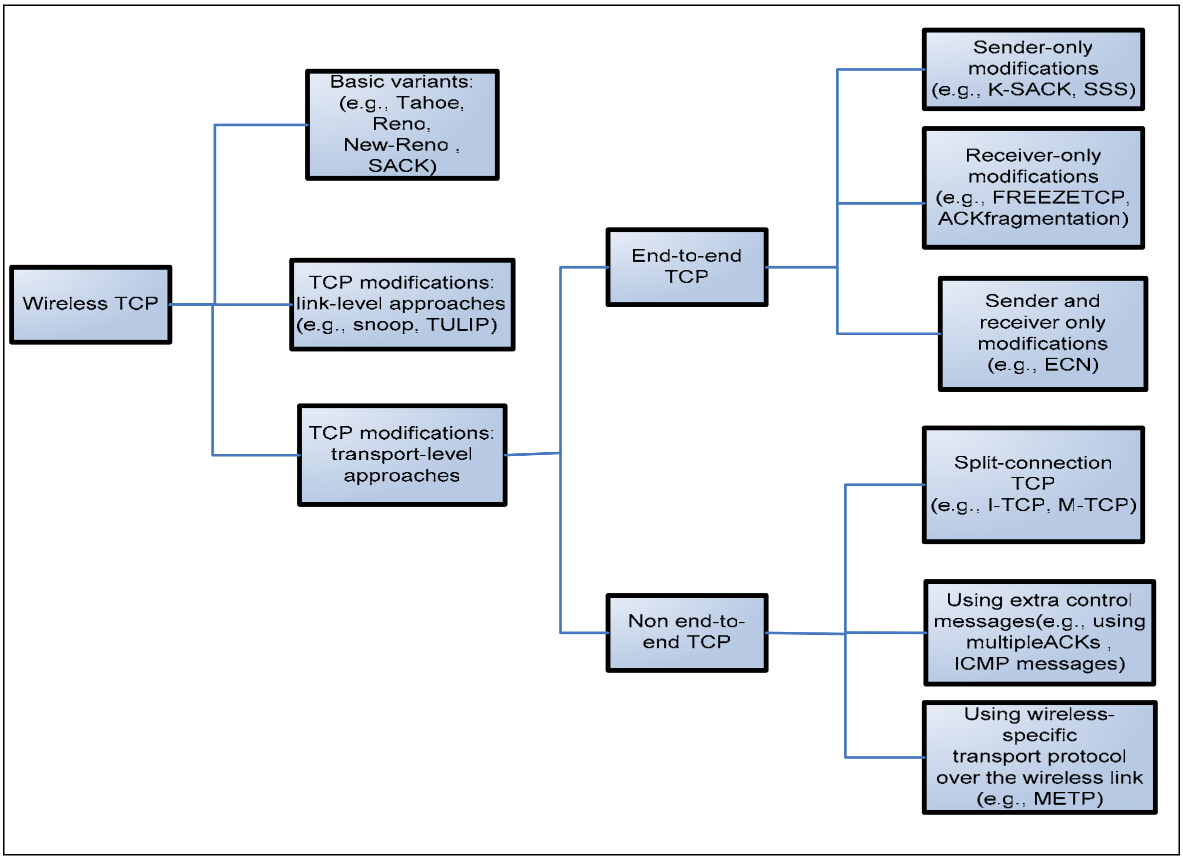

Wireless TCP
Basic variants: (e.g., Tahoe, Reno, New-Reno , SACK)
TCP modifications: link-level approaches (e.g., snoop, TULIP)
TCP modifications: transport-level approaches
End-to-end TCP
Non end-to-end TCP
Sender-only modifications (e.g., K-SACK, SSS)
Receiver-only modifications (e.g., FREEZETCP, ACKfragmentation)
Sender and receiver only modifications (e.g., ECN)
Split-connection TCP (e.g., I-TCP, M-TCP)
Using extra control messages(e.g., using multipleACKs , ICMP messages)
Using wireless-specific transport protocol over the wireless link (e.g., METP)

Figure 2.8 TCP solutions for wireless networks [37]

2.5.1 Snoop TCP [38]

The main function of the enhancement is to buffer data close to the mobile node to perform fast local transmission in case of packet loss. In this approach, the foreign agent buffers all packets with destination mobile node and additionally ‘snoops’ the packet flow in both directions to recognise acknowledgments [39, 40].

The reason for buffering packets toward the mobile node is to enable the foreign agent to perform a local retransmission in case of packet loss on the wireless link. The foreign agent buffers every packet until it receives an acknowledgment from the mobile node within a certain amount of time; either the packet or the acknowledgment has been lost. Alternatively, the foreign agent could receive a duplicate ACK which also shows the loss of a packet. Now the foreign agent retransmits the packet directly from the buffer, performing a much faster retransmission compared to the correspondent node.

The time out for acknowledgments can be much shorter, because it reflects only the delay of one hop plus processing time.

To remain transparent, the foreign agent must not acknowledge data to the correspondent node. This would make the correspondent host believe that the mobile host has received the data and would violate the end-to-end semantic in case of foreign agent failure. However, the foreign agent can filter the duplicate acknowledgments to avoid unnecessary retransmissions of data from the correspondent node. If the foreign agent now crashed, the time-out of the correspondent node still works and triggers a retransmission. The foreign agent may discard duplicates of packets already transmitted locally and acknowledged by the mobile node. This avoids unnecessary traffic on the wireless link.

Data transfer from the mobile node with destination correspondent node works as follows. The foreign agent snoops into the packet stream to detect gaps in the sequence number of TCP. As soon as the foreign agent detects a missing packet, it returns a negative acknowledgment (NACK) to the mobile node. The mobile node can now retransmit the missing packet immediately. Reordering of packets is done automatically at the correspondent node by TCP.

2.5.2 Indirect TCP [41]

Regular TCP is used from a fixed host to a base station whereas a modified TCP protocol that suits to wireless environment is used from the base station to a mobile node (MN). So, transmission errors on wireless link are not propagated to a wired network.

I-TCP segments a TCP connection into a fixed part and a wireless part. The correspondent node could also use wireless access. The following would then also be applied to the access link of the correspondent node.

Standard TCP is used between the fixed computer and the access point. No computer in the internet recognises any changes to TCP. Instead of the mobile node, the access point is now seen as the mobile node for the fixed host and as the fixed host for the mobile host. Between the access point and the mobile host, a special TCP, adapted to wireless links, is used. However, changing TCP for the wireless link is not a requirement. Even an unchanged TCP can benefit from the much shorter round trip time, starting retransmission much faster. A good place for segmenting the connection between mobile node and correspondent node is at the foreign agent for mobile IP. The foreign agent controls the mobility of the mobile node anyway and can also hand over the connection to the next foreign agent when the mobile node moves on.

The correspondent node in the fixed networks does not notice the wireless link or the segmentation of the connection. The foreign agent acts as a proxy and relays all data in both directions. If the correspondent node sends a packet, the foreign agent acknowledges this packet and tries to forward the packet to the mobile node. If the mobile node receives the packet, it acknowledges the packet. However, this acknowledgment is only used by the foreign agent. If a packet is lost on the wireless link due to a transmission error, the correspondent node would not notice this. In this case, the foreign agent tries to retransmit this packet locally to maintain reliable data transport.

Similarly, if the mobile node sends a packet, the foreign agent acknowledges this packet and tries to forward it to the correspondent node. If the packet is lost on the wireless link, the mobile nodes notice this much faster due to the lower round trip time and can directly retransmit the packet. Packet loss in the wired network is now handled by the foreign agent.

I-TCP requires several actions as soon as a handover takes place. The access point acts as a proxy buffering packets from retransmission. After the handover, the old proxy must forward buffered data to the new proxy because it has already acknowledged the data, after registration with the new foreign agent; this new foreign agent can inform the old one about its location to enable packet forwarding. Besides buffer content, the sockets of the proxy, too, must migrate to the new foreign agent located in the access point. The socket reflects the current state of the TCP connection, i.e., sequence number, addresses, ports etc. no new connection may be established for the mobile node, and the correspondent node must not see any changes in connection state.

2.5.3 Mobile TCP [42]

The M-TCP approach has the same goals as I-TCP and snoop TCP, to prevent the sender window from shrinking if bit errors or disconnection but not congestion cause current problems. M-TCP wants to improve overall throughput, to lower the delay, to maintain end-to-end semantics of TCP, and to provide a more efficient handover. Additionally, M-TCP is especially adapted to the problems arising from lengthy or frequent disconnections.

M-TCP splits the TCP connection into two parts as I-TCP does. An unmodified TCP is used on the standard host-supervisory host (SH) connection, while an optimised TCP is used on the SH-MH/MN connection. The supervisory host is responsible for exchanging data between both parts similar to proxy in I-TCP. The M-TCP approach assumes a relatively low bit error rate on the wireless link. Therefore, it does not perform caching/retransmission of data via the SH. If a packet is lost on the wireless link, it has to be retransmitted by the original sender. This maintains TCP end-to-end semantics. The SH monitors all packets sent to the MH and ACKs returned from the MH. If the SH does not receive an ACK for some time, it assumes that the MH is disconnected. It then chokes the sender by setting the sender's window size to 0. Setting the window size to 0 forces the sender to go into persistent mode, i.e., the state of the sender will not change no matter how long the receiver is disconnected. This means that the sender will not try to retransmit data. As soon as the SH (either the old SH or a new SH) detects connectivity again, it reopens the window of the sender to the old value. The sender can continue sending at full speed. This mechanism does not require changes to the sender's TCP.

The wireless side uses an adapted TCP that can recover from packet loss much faster. This modified TCP does not use slow start, thus, M-TCP needs a bandwidth manager to implement fair sharing over the wireless link.

2.5.4 Freeze TCP [43]

Freezing TCP is designed for longer interruptions of transmission. Examples are the use of mobile nodes in a car driving into a tunnel, which loses its connection to e.g., a satellite (however, many tunnels and subways provide connectivity via a mobile phone), or a user moving into a cell with no capacity left over. In this case, the mobile phone system will interrupt the connection. The reaction of TCP, even with the enhancements of above, would be a disconnection after a time out.

Quite often, the MAC layer has already noticed connection problems before the connection is actually interrupted from a TCP point of view. Additionally, the MAC layer knows the real reason for the interruption and does not assume congestion, as TCP would. The MAC layer can inform the TCP layer of an upcoming loss of connection or that the current interruption is not caused by congestion. TCP can now stop sending and 'freezes' the current state of its congestion window and further timers. If the MAC layer notices the upcoming interruption early enough, both the mobile and correspondent node can be informed. With a fast interruption of the wireless link, additional mechanisms in the access point are needed to inform the correspondent host of the reason for interruption. Otherwise, the correspondent node goes into slow start assuming congestion and finally breaks the connection.

As soon as the MAC layer detects connectivity again, it signals TCP that it can resume operation as exactly as the same point where it had been forced to stop. For TCP time simply does not advance, so no time expire.

The "Freeze-TCP" mechanism allows the mobile node to inform the sender for a possible disconnection. The goal of this approach is to enhance TCP performance over the wireless link with minimal packet loss and to avoid congestion control during the period of disconnection. This scheme requires only the mobile host's TCP code to be modified without any modification made to the intermediary i.e. to the base station or the sender.

During a handover or a possible disconnection, the mobile node will send out a few zero window advertisement (ZWA) with zero window size. When the sender receives a zero window advertisement, the sender will freeze all re-transmit timers, stop transmitting data and enter into persist mode. During persist mode, the sender will send probes called zero window probes (ZWPs) periodically to find out whether the window size of the receiver has been increased. When the window size of the receiver has been increased, the sender will send data at full speed and thus improve TCP throughput.

Since ZWPs are exponentially backed off, there is the possibility of substantial idle time after the reconnection. This could happen if the disconnection period is too long and reconnection happened immediately after losing a ZWP from the sender. To avoid this idle time, as soon as a connection is re-established, the receiver will leave persist mode and send three copies of ACK for the last data segment it received prior to the disconnection. This scheme is known as TR-ACKs (Triplicate Reconnection ACKs), [44]. In standard TCP, packet retransmissions are exponentially backed off therefore post reconnection idle time can occur as well. It is suggested by [43] to achieve a fair comparison, the standard TCP on the sender side be also modified to optionally send TR-ACKs. This way, the effect of only Freeze-TCP mechanism (i.e., forcing the sender into ZWP mode prior to a disconnection) can be isolated.

In the case of Freeze-TCP, since changes are restricted to the client end, holding back the ACK for the last byte does not help. Freeze-TCP will avoid any re-packetisation penalty at the sender end (which M-TCP might incur because it holds back the ACK to the last byte).

3 Chapter Three
Thin Client computing and Collaborative environments

This Chapter looks at the main concepts in computing that inspired the development of Virtual Network Computing (VNC) starting with thin client computing concepts. Afterwards it looks at the origins of VNC starting with X windows system and explaining the teleporting and Videotile concepts (both were developed at the Olivetti Research Laboratory responsible for developing VNC). I also look closely into the RFB protocol also known as the VNC protocol, which is the foundation of the design of VNC. Also, because of the fact that VNV is a collaborative environment, this chapter investigates collaborative environments and its characteristics in general and how to enhance the users' experience.

3.1 Thin client computing

In thin client computing, network application uses a form of communication known as the client-server paradigm. A server application waits passively for contact, while a client application initiates communication actively [45].

Thin client computing involves connecting thin client software or thin client hardware device with the server side using a highly efficient network protocol. Its architecture enables 100 percent server-based processing, management, development, and support for mission-critical, productivity, web-based, or other custom applications across any type of connection to any type of client hardware, regardless of platform. The client hardware can include window-based terminals, PCs, NetPCs, NCs, Apple Macintosh computer or UNIX devices.

In the client/server architecture the measurement of the fatness of the client is based on how much processing is done on the workstation. According to industry analyst, fat increases the *Total Cost of Ownership* (TCO) because of the maintenance required to keep the client workstation running [46]. In thin client environment all applications, processing power and user configuration reside on the attached server while the thin client will only display the remote applications, which are running on the server. VNC implements the concept of thin-client computing where applications run on a central server and simply deliver screen updates and inputs to clients.

Application service Provision (ASP) is a high profile implementation of thin client computing. The ASP model is the delivery of commercial software via the Internet or private leased lines for a fixed monthly subscription per user. Customers are able to benefit from lower costs and reduced management distraction as they are not required to own, maintain and upgrade their application software/hardware; or recruit and retain expensive in-house IT resource. In chapter 6 this feature is discussed in detail with relevance to VNC.

3.1.1 The advantages and drawbacks of thin client computing [47]

3.1.1.1 The Advantages of Thin client computing

- Reduction of costs -- The use of thin client computing can reduce the costs of upgrading hardware and software whereas in fat client, a PC usually has the life span of two years and it can be used as long as it survives.
- Reduction of bandwidth and power consumption -- One of the good advantages of thin clients is the saving of bandwidth where the bandwidth utilisation drops significantly. Thin client infrastructure has a very low bandwidth overhead. This is due to the fact that each user sends keyboard

strokes and mouse movements to the server and receives the corresponding changes to the display. In addition to that, the reduction in the power consumption of the thin client compared to the PC. This is explained in [48] where Steve Greenberg shows his results in his paper.

- Speed of deployment, repair and replacement -- In a PC environment software must be updated on a regular basis which might take time to IT staff to reach remote servers to upgrade them which is unacceptable for security and virus updates. In thin client computing, updates are done centrally on the server where all clients have access to it with its up-to-date versions of all software.

3.1.1.2 Drawbacks of thin client computing

- Users nowadays accepted PCs and got up close and personal with them even in their work where they familiarise their computing environment by changing screen savers and customising it. In addition to that they are used to their floppy disks and CD-ROMs yet these options are disabled which may lead to the refusal of adopting dumb terminals, however, using a tubby client is a middle solution that suits everyone. Tubby client are defined in [45] as " PCs that have operating systems and some applications installed on them, however it uses a local installed thin client to connect to their thin computing environment for many applications".
- Another disadvantage is clear in high performance environment where high rate of keyboard and mouse refreshes are needed which therefore leads to a high load of network traffic. This is the case in CAD and DTP environments, so a fat-client model is a better choice.

3.2 The X Window System [49]

X is a portable, network-transparent window system using a client/server model. The X window system runs on many different computers. Popular platforms include workstations from companies such as: Sun Microsystem Inc, Silicon Graphics Inc and IBM PCs running Linux. There have been several versions (releases) of X window system. However, it was not until the eleventh version known as X11, which it was widely released and began the popularity it occupies today. It is worth mentioning that the naming of X client and X server is the reverse of the traditional client/server architecture. Every X window system is based on three parts as illustrated in Figure 3.1 which are the X server and the X client and the X protocol that connects between them. The X Windows system makes it difficult to migrate the window of a running application from one screen to another.

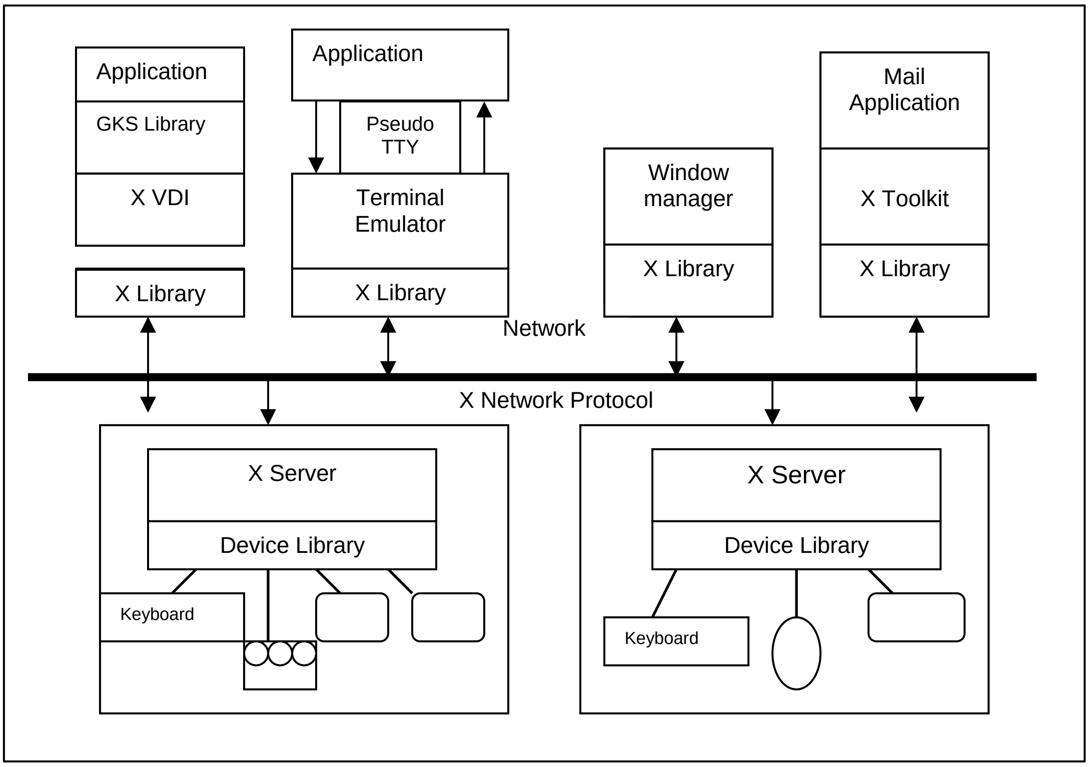


Figure 3.1 X window system block diagram [50]

3.2.1 The X Server

The X server is software that runs on the local machine and has the role of managing a single screen, keyboard and mouse. It accepts and demultiplexes network based X client requests and act upon them. The X server therefore is used to display drawing requests on the screen and to manage the keyboard, mouse and display device by multiplexing keyboard and mouse inputs onto the network to the respective X client. In addition to that the X server replies to information requests and reports any error in a request. X servers control what appears on the screen of a display by means of a directly writeable portion of video memory called a framebuffer.

3.2.2 The X Client

The second major component of any X window system is the X client, which is an application that is displayed on the X server and it typically runs on a remote machine which has excess computing power. X client is written with aid of libraries (Xlib, xt) and it sends requests to the server and receives events and errors from the server. X clients communicate with the server using the X protocol over a reliable byte-stream such as that provided by TCP/IP. A computer may have a number of displays, each with its own X server where the connection from X clients is accepted by the servers on their own well-known TCP port.

3.2.3 The X session

The X server offers a root window where the windows of clients are arranged. An x session starts when a user logs in and starts an X client; the session consists of many clients including the window manager, which is responsible for arranging the windows of other clients on the root window, and manipulates their windows.

The window manager is the component, which controls the appearance of windows and provides the means by which the user can interact with them. A large number of windows managers have been developed where they vary in appearance and behaviour and where they give you the ability to customise them and configure them to suit your wishes. More recent developments have created a new breed of desktop environments, which aim to provide a more complete interface and supply their own integration utilities and applications like freely available (KDE and GNOME) [51].

3.2.4 The X Protocol

The X protocol distributes the processing of information by specifying a client/server relationship between the application (X client) and its display (X server) at the application level. The X protocol consists of several types of messages sent between the client and server, such as Requests where X client requests a certain action from the X server, Replies where X server responds to the X client, Events where X server forwards an event to the application (for example a key or mouse is pressed) and Errors when an error is encountered in X clients request the X server will report errors.

3.3 The Teleporting system

There are two approaches to provide a mobile windowing environment.

The first approach is developing new mobile applications, which implement the mechanics of redirecting their input and output between different displays. The second approach is introducing a level of indirection between the client and the server to use existing applications that are unaware of mobility, this method employ a proxy server. Each approach has its advantages and disadvantages where they are mentioned in detail in [9].

The teleporting system was developed as a mean of experiencing mobile applications, without the need to rewrite existing applications or design new ones. It provides a mechanism for transparently introducing the mobility element into existing and familiar applications [12]. Teleporting is an attempt to achieve ubiquitous, personalised computing environment and also a mean of causing the application's output to disappear from one display and re-materialise on another [9].

The proxy server approach was chosen to develop the teleporting system due to its practicality and suitability to allow unmodified X clients to be used in a teleporting environment. In [52], In order to make existing applications mobile without attention is to use a proxy server (pseudoserver) between the client and the server where this introduces a level of indirection. Some proxy servers were written to display the output of applications on multiple displays like [53] but for the purpose of mobility new proxy servers had been developed to achieve application's mobility such as tpproxy [9]. The teleporting implementation is specific to the X window systems; Figure 3.2 outlines the structure of the teleporting system.

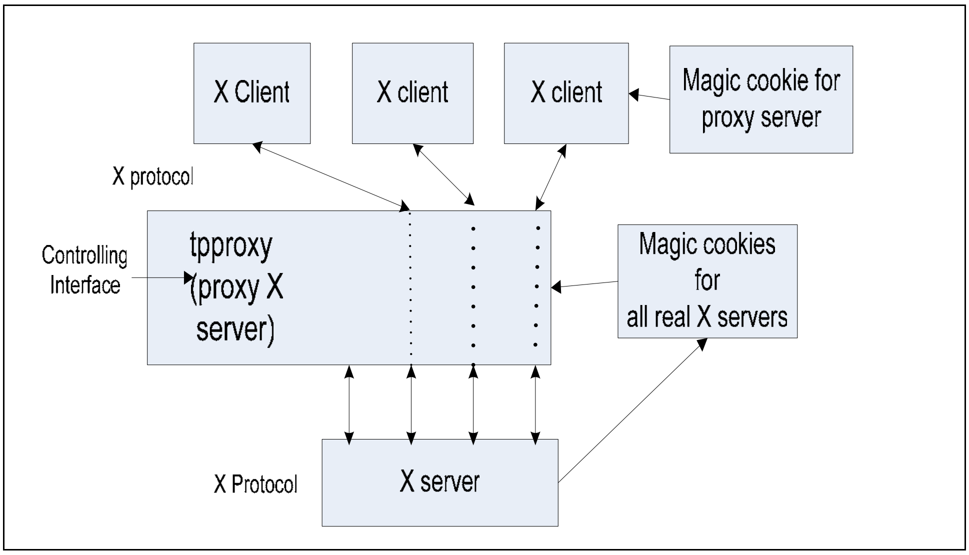


Figure 3.2 The Teleporting system structure [9]

At the centre of the figure is the proxy server *tpproxy* which accepts connection from X clients appearing to them as a real X server. When the proxy server moves its root window from the display of one X server to that of another *Teleporting* takes place.

The proxy server acts as a bi-directional filter where it receives requests from clients and translates them before passing them to the X server. In the reverse direction, it takes replies, events and errors from X server and translates them then passes them to the clients. Moreover the proxy server records all changes to the state inside the server relevant to its clients by keeping track of all the resources created by the clients and any modifications made to them. When teleporting occurs the proxy server should connect to the new server, replacing the current connection. The proxy server should re-create each client's resources and set their attribute in the right order consecutively to bring the new server to state of recognising that. A proxy server does not have a screen, keyboard or mouse (a display) of its own. Instead it is able to make use of the display of some real X server.

3.4 The Video tile [54]

Originally at ORL in 1994, they developed the RFB service for a video tile [54] which is a pen-based ATM-connected display where the RFB client running on the tile passes pen events as mouse events to the RFB X server and puts all screen changes it receives onto the tile display, thereby allowing interaction with X applications on the tile. The tile has a backlight colour active matrix liquid crystal display with a resolution of 640 by 480 pixels. Each pixel can display 512 colours. Because the VNC viewer is a software-only version of the Videotile, and so provides 'workstations' which can be created or deleted as well, the system was named **Virtual Network Computing**.

The videotile runs the ATMos operating system, rather than porting an X server or other windowing system to run on ATMos, a RFB protocol is used to display and interact with applications on the tile. The applications actually run on a server machine somewhere across the ATM network.

When running multimedia, video streams are needed to appear inside application windows. It would be extremely inefficient for a video stream from a camera to pass through a workstation before being displayed on the tile. Instead video is sent directly to the tile and at the tile the application's user interface (sent via the RFB protocol) is mixed with the video stream so that the video appears in the correct position on the screen and is clipped as appropriate. The tile has a pen interface for interacting with applications; in addition to raw pen input the tile can also use handwriting recognition. As with the windowing system, the handwriting recognition engine runs not on the tile itself but on a server machine somewhere else on the network.

3.5 Virtual Network Computing (VNC) [55]

For the usage of the teleporting system developed at Olivetti Research Ltd / AT&T Labs Cambridge, on a global scale, several requirements are needed such as a network and common interface widely available all over the world. The World Wide Web and Java provide such an infrastructure.

The implementation of such an arrangement is developed at ORL and is called (Virtual Network Computing). It was initially an attempt at global teleporting and was based on the idea that a work session is identified with a web page containing a Java applet. Windows and graphical applications in the VNC session are brought up in the browser and can be manipulated and it is possible to browse other pages returning to the VNC page whenever the user wants. In addition to that VNC provides the capability to disconnect a VNC session from one browser and either to call it up to appear in a new browser or to terminate the session to be called up at any later time. This implementation was originally called JavaTel (for Teleporting in Java) but was changed to VNC to avoid confusion with Java Telephony applications [55].

In order to move the concept of teleporting to a wider arena of the Internet, another sort of proxy server is used in VNC and was developed for UNIX workstations; it is called Remote Frame Buffer (RFB) service that is provided by an RFB X server where two features were added to the standard X server

1) It provides a TCP socket interface on which it will accept mouse and keyboard events.
2) It also provides another TCP socket interface where it will accept requests for information about the state of the screen display. As a reply to such request the RFB X server sends details of the changed regions of the screen in the form of a set of bitmapped rectangles.

The protocols used on the two sockets interface comprise the ***RFB protocol*** shown in Figure 3.3

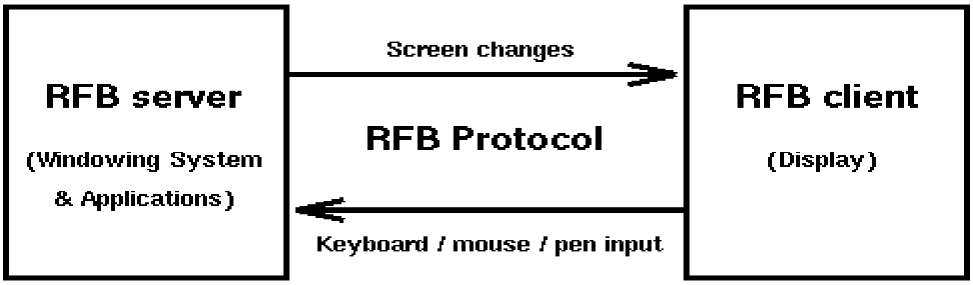

RFB server
(Windowing System & Applications)
Screen changes
RFB Protocol
Keyboard / mouse / pen input
RFB client
(Display)

Figure 3.3 The RFB protocol [54]

By connecting to those socket interfaces, an RFB client running on a remote (non-X-aware) device can provide seamless interaction between a user and the X server. “The RFB protocol is far simpler than X therefore it places very little demand on the remote display in terms of processing power, memory...etc”[54].

RFB is a simple protocol for remote access to graphical user interface. It is applicable to all windowing systems and applications including X11, Windows3.1/9x/NT/XP and Macintosh, because it works at the frame buffer level.

This protocol is a "Thin Client" protocol where only few requirements are needed from the client thus it can run on any kind of hardware.

The endpoint where the changes to the frame buffer originate is known the RFB server. The display side of the protocol is based on a single graphic primitive: "put a rectangle of pixel data at a given (x, y) position" by using this technique and allowing various different encoding for the pixel data a large degree of flexibility is achieved in how to trade off various factors such as network bandwidth, client drawing speed and server processing speed.

By writing an RFB client in Java, the ability to interact with an X server from within a Java applet can be provided. The RFB client applet opens sockets to the mouse/keyboard and screen-change ports of the RFB server. As a result, it sends the mouse and keyboard events from Java's AWT down the appropriate socket and paints changed rectangles to the screen as they arrive from the RFB server using the AWT. The result is the appearance of an X session within a web browser.

3.5.1 The structure of the VNC system for UNIX [55]

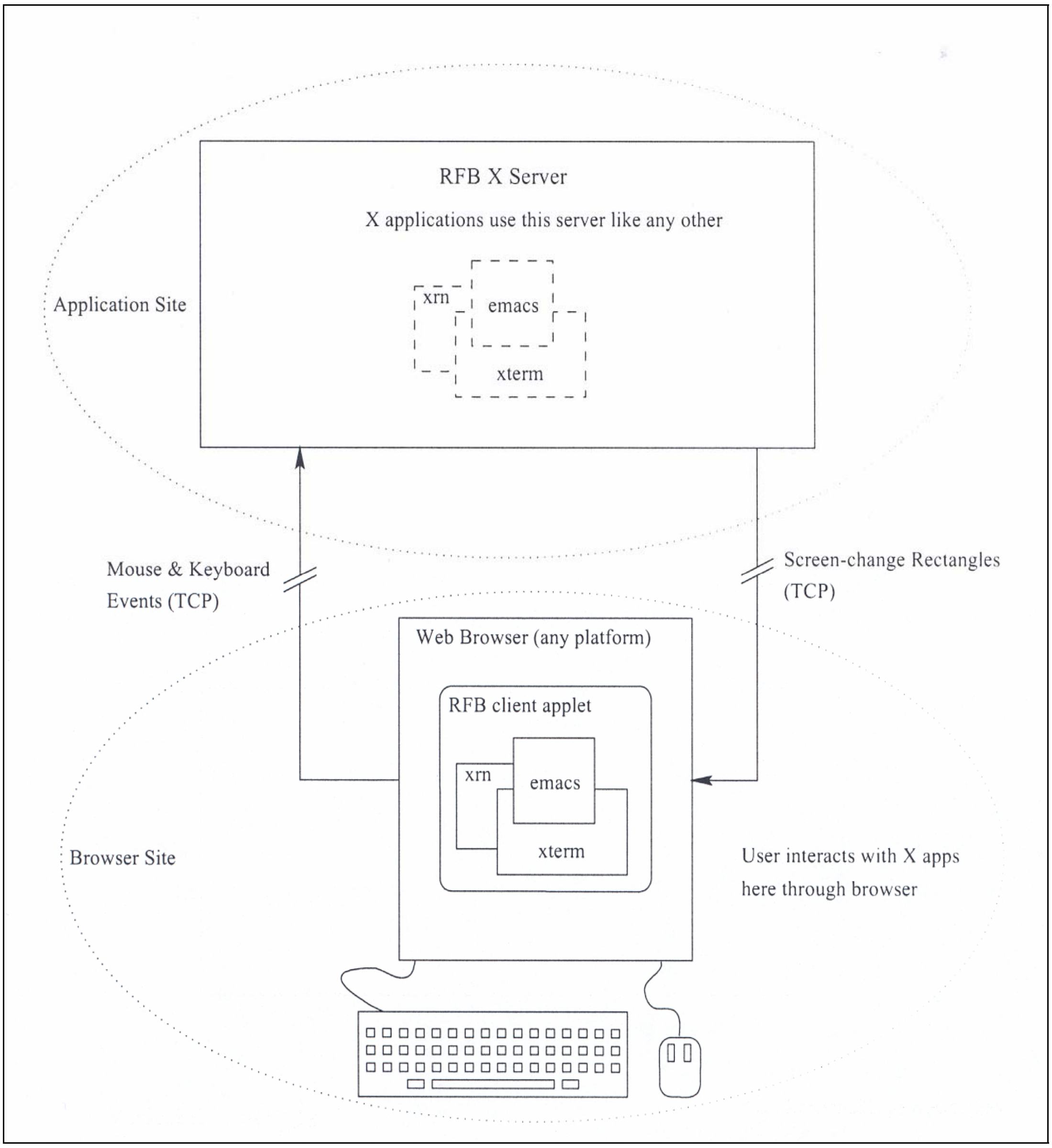

RFB X Server
X applications use this server like any other
xrn
emacs
xterm
Application Site
Mouse & Keyboard Events (TCP)
Screen-change Rectangles (TCP)
Web Browser (any platform)
RFB client applet
xrn
emacs
xterm
Browser Site
User interacts with X apps here through browser

Figure 3.4 Structure of VNC system [55]

Figure 3.4 represents the state of a connected VNC session where:

1) On the application site, the user had initiated his VNC session earlier by running a startup script.

This script:

a) Starts the RFB X server with its associated session.

b) Creates an HTML file, which encapsulates the corresponding RFB client applet.

2) On the browser site, the user points at the browser at the URL of the generated HTML file to download and run the RFB client applet. The HTML file must be served by an HTTP server running on the same machine as the RFB X server because of current security restrictions.

As mentioned in [55] VNC is a conceptual but not a technical extension of teleporting. The major difference between the teleporting and VNC system is that VNC client is much thinner than the teleporting client, therefore the client machine where the user brings up a teleporting session must run a full X server whereas in a VNC session the client machine is required to run only a very small Java applet (20K).

It is worth mentioning that XVNC server was developed before adopting VNC to work on Microsoft Windows. Creating the Windows VNC server according to [56] was much harder than the UNIX VNC server because Windows have fewer places to insert hooks into the system to monitor display updates, and the model of multi-user operation is less clearly defined, the current server simply mirrors the real display to a remote client, which means that only a single VNC desktop is available from any one PC.

The simplicity of the VNC protocol allows it to be applied to a wide range of hardware devices and different operating systems ranging from UNIX to Windows and Macintosh. VNC servers can also be programmed to run on any hardware devices other than computer workstations, e.g. on CD players, telephone answering machines, etc. [56, 57].

3.5.2 The RFB Protocol (The VNC Protocol) [58]

The RFB protocol [58] as mentioned before is a simple protocol for sending graphics to be displayed on a remote display where RFB stands for Remote Frame Buffer. RFB protocol sends the whole framebuffer from the server to the client but the current modified version detects the regions of changes of the screen and only sends them to the client.

The RFB protocol was originally developed as a mean of interacting with applications from an ATM network display device called Videotile (pen-based ATM connected display) and by writing a RFB client in Java the same can be achieved through a browser. It is possible to hook the RFB server into a workstation's normal display. The RFB server has a "Virtual framebuffer" and the RFB client is the only physical screen showing the contents of the framebuffer.

As mentioned before, RFB is "thin client" protocol where the design of the RFB protocol ensures to make few requirements of the client so it can run on a wide range of hardware. This protocol makes the client stateless where if the client disconnects from the server and then reconnects again to it, the state is preserved. The graphical user interface environment is the same even when different users connect to the server he/she will view the same graphical environment.

The RFB Protocol described here is VNC 3.3.7 however it has been modified on August 2003 and again on July 2005 for the new version of VNC 4.x.

3.5.2.1 The Display protocol

The graphics primitive of the protocol on the display side: "put a rectangle of pixel data of a given x, y position" and various encodings for the pixel data is allowed. A *frame buffer update* is a sequence of rectangles framebuffer and an update represents a change from one state to another. The client requests the update protocol where it's only sent from the server to the client as a response to a request from the client.

3.5.2.2 The Input protocol

The input side of the protocol is based on a standard workstation model of a keyboard and a pointing device. Input events are sent from the client to the server whenever a key is pressed or when the pointing device is moved.

Initially when the RFB server interacts with the RFB client they negotiate the *format*[1] and *encoding*[2] of the pixels to be sent between both of them. The main design consideration is to leave most of the job on the server that should have the ability to supply the suitable format to the client. Before sending the data a header is sent containing the: x, y position of the rectangle of pixels, the width and height of the rectangle and the encoding type. The RFB protocol can run on any reliable transport either on byte-stream or message-based.

Figure 3.5 outlines the process of establishing a VNC session, starting from when the VNC viewer connects to the VNC server to establishing the connection stream and then exchanging messages between the client and the server.

[1] Is individual colours represented by pixel values.
[2] Is the way a rectangle of pixels is sent on the wire.

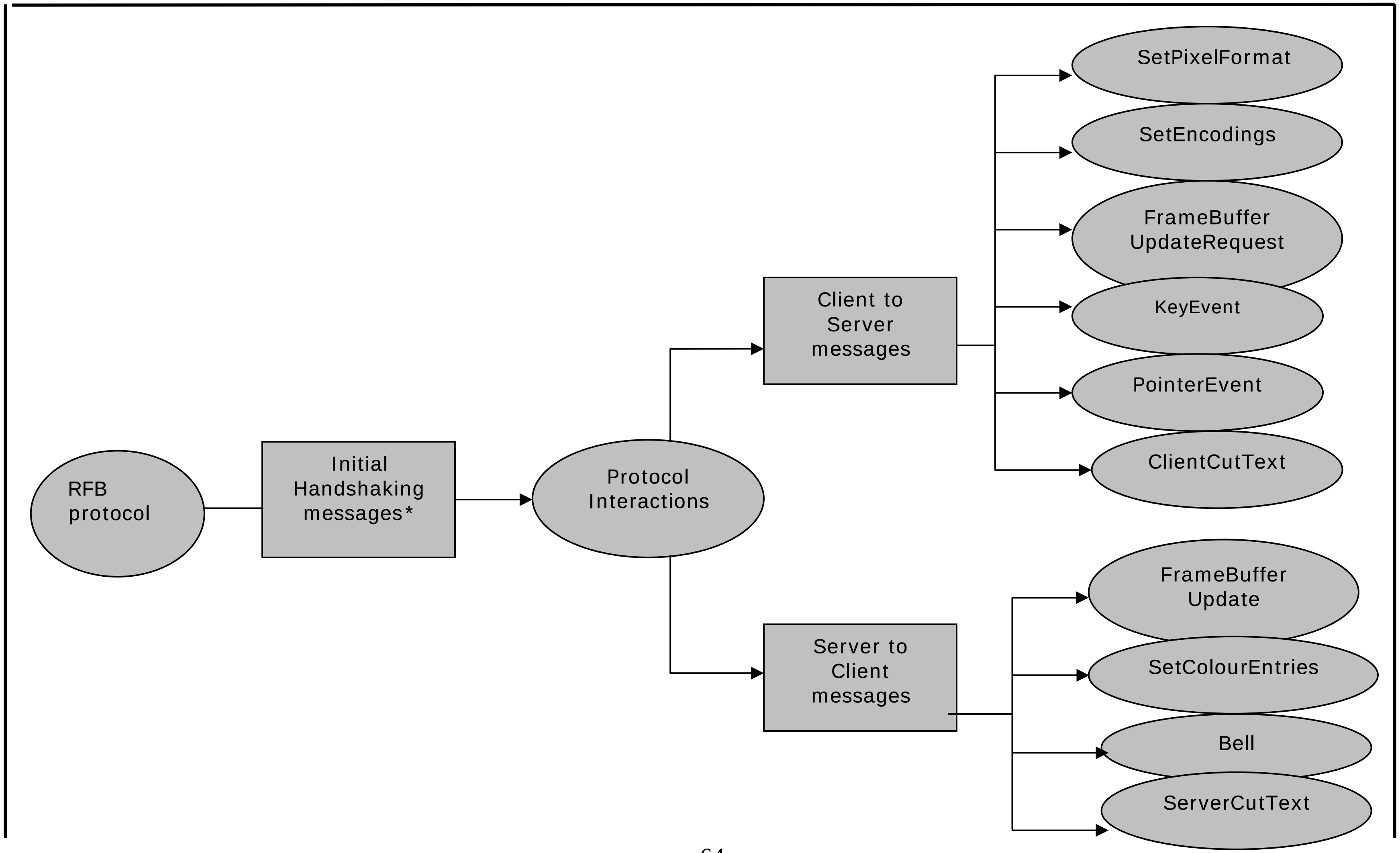


Figure 3.5 An outline of the RFB protocol mechanism (* in more detail in figure 3.6 and 3.7)

3.5.2.3 Initial Handshaking process

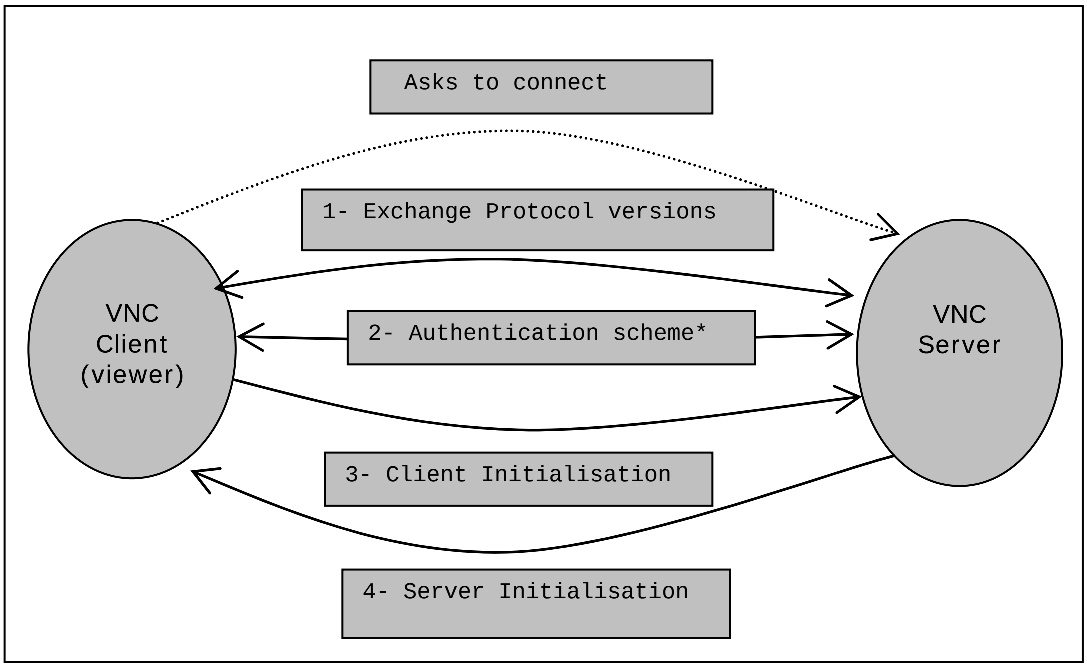


Figure 3.6 Initial Handshaking (* in more detail in figure 3.7)

Looking at Figure 3.6, the protocol starts from the top with the process of handshaking between the client and server where

Firstly, a protocol version message is sent from the server to the client informing with the latest RFB protocol version number it supports. Then, the client sends a reply giving the version number it wants to use. Servers are responsible of the version compatibility and could adapt to different scenarios of versions in compatibility.

Secondly, authentication: after that the server sends a message indicating the authentication method to be used that can be either:

1- Connection failed: the server cannot support the version number.
2- No authentication: when no password is needed.
3- VNC authentication: where VNC password is needed and the server sends a 16-byte challenge, the client must encrypt it with DES using a key and sends the

16-byte response. Now, the server sends back to the client a word to indicate the status of the connection where it is either:

A) OK: authentication successful.

B) Failed.

C) Too-many: where too-many failures of authentication to the same client took place and therefore it disallows immediate connection by this specific client (maximum of 6 attempts by default).

Stage two is explained in more details in Figure 3.7

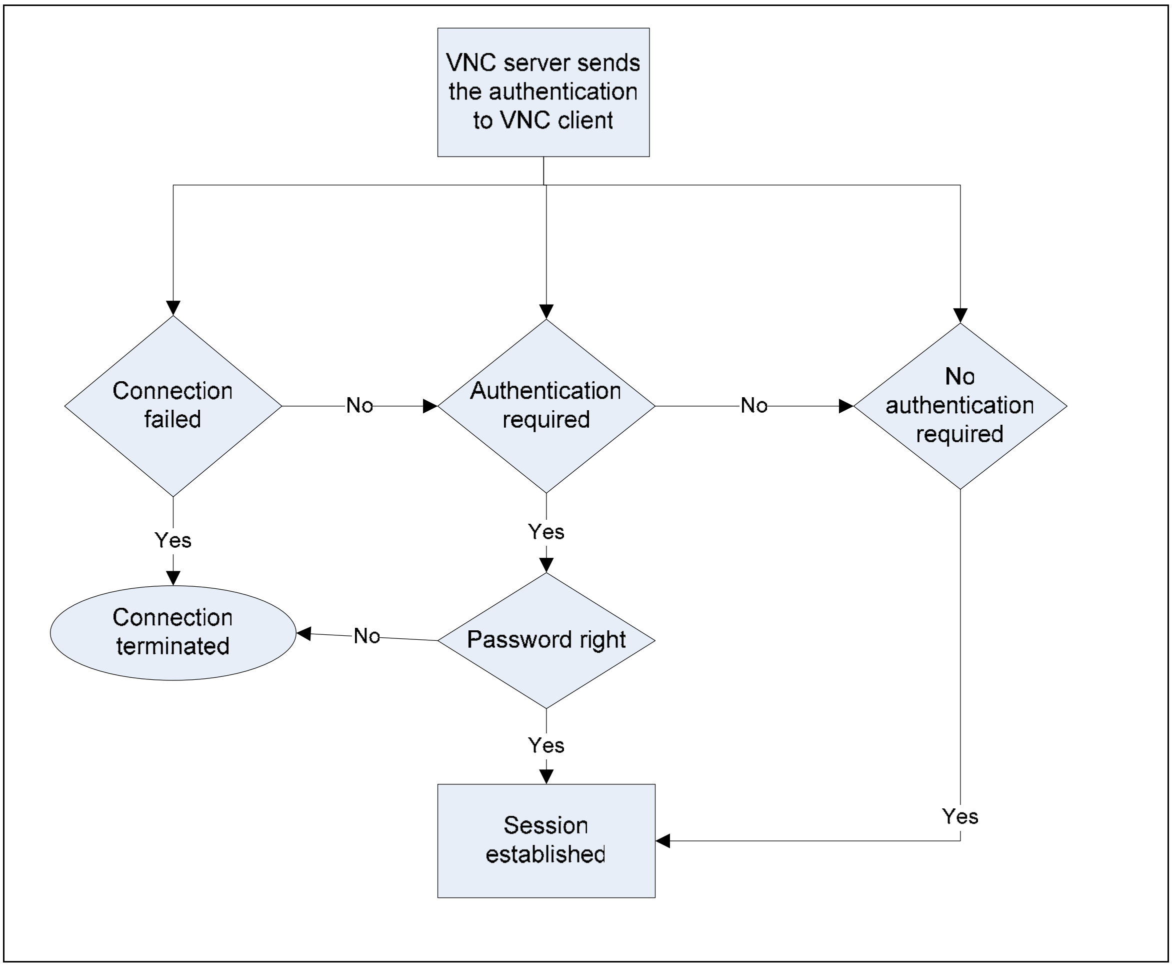


Figure 3.7 Authentication scheme

Thirdly, client Initialisation: after establishing the trust between the client and server, the client sends a message either by asking for a *shared* connection where other clients can still be connected or *non-shared* flag where all current connection must be disconnected.

Fourthly, Server Initialisation: where the server sends to the client the following information:

1- Framebuffer width
2- Framebuffer height
3- Server pixel format
4- Desktop name

3.5.2.4 RFB Protocol interactions

A. Client to server messages

- SetPixelFormat: this message overrides the pixelformat in the serverInitialisation message. If cannot be used the server's pixels format is used. A true colour map must be used if the true-colour flag is zero.
- SetEncodings: where it sets the encodings types it would like the pixel data sent as and then it lists its preference according to the most preferred.

  Types of encoding:

  Raw encoding, CopyRect encoding, RRE encoding, CoRRE encoding, Hextile encoding, ZRLE encoding, explained in detail in [57].
- FrameBufferUpdateRequest: where the client sends to the server a notification of the (x-position, y-position, width, height) of the area needed.

1- If the client has a copy of all parts of the framebuffer that it demands, then the server sends incremental updates to the client.

2- Otherwise, it sends the FrameBufferUpdateRequest with incremental set to zero so that the server must send the whole content of the area demanded.

The reply to this message is the *FrameBufferUpdate* from the server.

- KeyEvent: a key press or release using the down-flag = 1 : pressed
  =0 : released

  The key is specified using the "keysys" values defined by the X windows system.
- PointerEvent: pointer movement (x-y position), pointer button press or release.
- ClientCutText: new ASCII text in its cut buffer.

B. Server to client messages

- FrameBufferUpdate: a response to the FrameBufferUpdateRequest from the client. And one FrameBufferUpdate can respond to multiple FrameBuferUpdateRequests.
- SetColourMapEntries: if "a colour map" to be used. This message specifies to the client the pixel values mapped to the given RGB intensities.
- Bell: a bell rings on the client if it has one.
- ServerCutText: new ASCII text in its cut buffer.

3.6 Collaborative environments and Groupware

Computer Supported Cooperative Work (CSCW) refers to the field of study that examines the design, adaptation and use of groupware. Groupware or also known as Collaborative Environments (CE) is a technology designed to facilitate the work of groups. This technology is used to facilitate communication and make it faster; clearer and more persuasive. Also it is used to enable communication where it would not otherwise be possible; moreover it enables telecommuting and cut down the travel costs and brings together multiple perspectives and expertise. Groupware is also used to save time and cost in coordinating group work and to facilitate group problem-solving and to enable new modes of communication.

Desktop conferencing, videoconferencing, co-authoring features and applications, electronic mail and bulletin boards, meeting support systems, voice applications, workflow systems, and group calendars are key examples of groupware. Labels vary: groupware, collaborative environments, workgroup computing, multi-user applications, and computer-supported cooperative work (CSCW) applications.

Geographically distant researchers have the choice to use many developed CSCW applications such as Whiteboards Tools and the Shared Application Tools such as Quilt [59] and GRoup Outline and Viewing Editor (GROVE) [60], which implement What You See Is What I See (WYSIWIS) mode. For more groupware tools refer to [61]. These applications implement What-You-See-Is-What-I-See (WYSIWIS) [62] mode where remote participants can view and be involved in common tasks. WYSIWIS maintains the illusion of one interface to a shared artefact by mapping all user-initiated interactions from all users to equal input/output interactions at all replica interfaces as soon as possible, with the same ordering.

One of the greatest successes in the field of CSCW has been the introduction of shared workspaces. A shared workspace provides a virtual place to work, tools for performing the work, and channels for communication among its inhabitants. At the same time, a shared workspace does not compromise the flexibility in the support it provides; a shared workspace can be used for a variety of tasks. Because of this flexibility, many groupware systems use shared workspaces as one main basic component for supporting cooperation. Shared workspace also refers to the remote sharing of computer display work surfaces between participating individuals involved in common tasks and who collaborate without leaving their workplaces [63].

The common properties shared workspace applications have are: firstly, there is at least one portion of computer display that is viewed by all participants in a WYSIWIS manner. Secondly, they involve a desktop computer or X-terminal except for electronic whiteboards also known as Liveboards. Thirdly, they are dedicated to the professional CSCW environment. Fourthly, they are rarely used alone, but simultaneously with at least audio and sometimes video direct communications to provide a complete and integrated remote and synchronous cooperative work environment. Shared workspace applications aim to create at a distance the computer-mediated sharing of ephemeral information as well as sharing the actual products on which participating individuals are working [63].

The essence of groupware is a technology that supports *communication* between participants in synchronous or asynchronous manner; it should also support *collaboration* or *cooperation* in a shared information space where collaborative work generally involves creation of some artefact representing the outcome, and this can be done in shared-information-space that provide virtual places where people create and manipulate information (real time and asynchronous). Groupware technology also supports *coordination* of the collective contribution where coordination features facilitate interaction between or among participants, Coordination features are essential when interacting asynchronously in shared information space where access control features limit who can participate in a shared space. [64]

3.6.1 Characteristics of Collaborative environments

Collaborative applications coordinate activities which may be distributed in time and/or space. Distribution in time means that activities may take place at different times, but are coordinated to achieve a unified effect (such as the production of a document). Distribution in space means that activities may take place on different computers, perhaps linked by a data network.

There are variable factors that construct a unique Collaborative environment from one session to another [65, 66] such as

- Group structure and social roles like (group size, grouping of abilities, age, gender and background), and where in [65] a group writing taxonomy is defined comprising the social roles of writer, consultant, editor and reviewer.
- Task type which is the tasks that people perform when they are collaborating which stretches from the conception of an idea up to the moment of dissemination and it involves a broad range of activities such as brainstorming, planning, researching, writing, reviewing, and editing in different settings such as meetings, collaborative work, education, presence and entertainment .
- Task requirement which is the requirements the users need as they perform the tasks such auditory communication, visual communication, audio/video synchronisation, video conferencing, a shared workspace, turn-taking ability. Understanding the task, and therefore the requirements of that task is important for delivering a high quality collaboration experience using the appropriate technology.

Moreover, a number of criteria are required to be met in collaborative applications in order to present users with a single uniform data space across a network according to [67 , 68, 69] such as

- *availability*—users should be able to gain access to data when they need it;
- *transparency*— users should not have to worry about patterns of data distribution, or the details of the distribution management;
- *consistency*—users should see identical (or, at least, consistent) views of shared data, even though they may be working at different places or different times;
- *Responsiveness*— data management should not interfere with the interactive response of the system;
- *Security and Privacy*—providing security and privacy to the session's users by using session control policies such as avoiding intrusive situations where users are able to invade privacy or impose a session on others and providing a means for preventing interruptions. Also deciding how much information to share

about the users is an issue to consider because anonymity can be crucial in encouraging fair participation in discussions and is useful for providing protection from harassment ; and

- *Asynchronous collaboration* — In order to support global cooperative work, the system being developed must concentrate not only on synchronous collaboration but also on asynchronous collaboration. Synchronous collaboration is where participants in the CSCW session are available simultaneously where unavailable participants can then join the session later in an asynchronous manner to view what they missed in the session.

However, these criteria place conflicting demands on an implementation. For example, increasing availability can affect consistency in the sense if we enhanced availability by maintaining multiple copies of the data on different network nodes means that consistency is affected because two users can make incompatible changes to two copies of the same piece of data.

When trying to resolve these conflicts, transparency can be endangered by introducing more ways in which users can be exposed to the consequences and details of distributed data management. Therefore it all depends on the requirements of the application which differ from one application to the other.

Furthermore, when security is compromised availability can be affected, for example is that Denial of service (DoS) attacks causes service unavailability despite the fact that the underlying transport infrastructure is fully capable of offering the requested resources. On the other hand tightening security measures can affect availability too. For example, enabling to limit who can join a session can conflict with availability depending on the criteria used such as in the case if a legit user who is permitted to join the session is not allowed in because session control limited the number of users allowed to join.

Therefore, one important point that we should consider is related to availability is the data distribution issue and where a particular data structure will reside in the system at any given time. Two approaches are centralisation and replication. In the centralisation approach, such as in thin clients, the data is kept at one point in the system, in this case consistency is a trivial issue since there is only one copy of data. However, replication allows multiple copies of data structures which improves availability, but complicates consistency management.

Another important point that we should consider is related to consistency and how to maintain data management and consistency in the face of the simultaneous activity of multiple users. Inconsistency generally arises through disordering in applying individual changes to user data at different sites. User actions arise independently at different points in the network, and are then propagated to other users. This distributed activity introduces timing problems; event notifications may arrive at different nodes in different, unpredictable sequences. To maintain consistency, the system must ensure that each client sees the result of these changes applied *in a consistent order*.

3.6.1.1 Consistency Guarantees

The easiest way to manage and avoid inconsistencies is by providing users with consistency guarantees. Such guarantees are "promises" that some form of consistency can be maintained within the shared document in the presence of particular concurrent interactions.

Two approaches from the area of (distributed) database management have been used in collaborative systems [70]**.** These approaches give rise to two different types of observable behaviour for users: serialisation and locking.

- *Serialisation* refers to a particular scheduling of interactions and involves *postponing* particular interactions. This is done in order to enforce a particular non-conflicting ordering on the feedback and feedthrough of related interactions.
- *Locking*, on the other hand, refers to giving privileged access to some user upon request and involves *denying* particular interactions of other users on particular shared objects to which a user has acquired a lock.

Locking can also be referred to as floor control policies. These policies regulate access to the shared objects to preserve social rules and protocols that govern the access to the shared space. Many common floor control policies can be regarded as locks on the entire workspace, restricting activity to one individual at a time. This is *input multiplexing*—the reduction of multiple input channels (one or more per individual collaborator) to a single channel (the input channel to the workspace). Essentially, that participant holds a lock on the entire workspace; no other participant can contribute until she loses control (relinquishes her "lock"), and so consistency is maintained.

However, in general, there are four floor control policies: No control, implicit locking, explicit locking, and chair control.

First, No control: this approach gives every participant the freedom to access the shared workspace and it depends on common sense and rules of social behaviour to resolve any conflict. Second, implicit locking: when a participant starts typing information he/she implicitly takes the floor and no one else can enter any data. There is a specific time for the user before his holding the floor come to an end. Third, explicit locking: similar to the above, except the user must explicitly request the floor via a dedicated key or click. Fourth, chair control: a chairperson (moderator) is designated, he/she have the power to hand over the session to who requests it and regain it at any time. This approach needs a tool to recognise all the pending floor requests.

3.6.1.2 Workspace Awareness

In general, Workspace awareness comes naturally in a face-to-face situation, but it is far more difficult to maintain in a real-time groupware system. In groupware, people may only see a fraction of the workspace, and may not see the same part as other group members. A groupware system also reduces the richness of communication, and its interface may hide many actions that are visible in a physical workspace. Furthermore, perceptual and physical abilities that we use to maintain workspace awareness (such as glances) are often replaced with mechanisms that are comparatively slow and clumsy (such as scrolling). [71]

Within this different environment, the groupware designer must try and recreate the conditions and cues that allow people to keep up a sense of workspace awareness. *workspace awareness* is the collection of up-to-the minute knowledge a person uses to capture another's interaction with the workspace and the condition in which a group member perceives the presence of the others and where persons with whom a communication episode can be initiated. In closely coupled visual tasks, people use the representation of the other person as a kind of visual evidence of the state of the action [72, 73]. In the real world, people quickly learn to use this information to improve their efficiency. They can become expert at a shared task by learning the task boundaries that are presented by the other person, and gradually pushing towards those limits. This expertise, however, depends upon the information being accurate.

It is worthy to pinpoint to one of the most useful awareness elements of real-time groupware, telepointers. They are simple and computationally inexpensive, but provide embodiment, awareness, and gestural communication. Therefore using them would reduce the likelihood of divergence, as users are more aware of other users' interactions. Observing others approaching an object with a pointer or changing a particular sentence makes conflicting interactions "impolite" within the social protocol between users.

However, telepointers often suffer from severe performance problems on real-world networks like the Internet. When the network becomes congested, telepointers become jumpy and slow, often to the point where they are no longer useful to the collaboration. In situations where people use telepointers to coordinate closely-coupled interactions, these incorrect representations of the other person's actions can lead to frustration and errors in the collaborative activity.

The problem is that although telepointers can convey a great deal of information, that information is sensitive to issues of lag, synchronization, and pacing. Disruptions to these qualities are caused by network latency, jitter, and loss – all of which happen frequently, even on high-bandwidth networks. As a result, telepointers are virtually unusable in groupware that operates on real-world networks: most common groupware applications do not even attempt to provide telepointers (e.g. MSN Messenger whiteboard, NetMeeting, Groove, and nearly all multi-player games); screen-sharing systems (e.g. VNC, Citrix) that do provide a single shared cursor suffer clear performance problems when network difficulties arise and the case that many users are moving to lower-bandwidth and more error-prone connections such as cable, ADSL, and wireless networks. Yet the problem will not soon be solved by increasing network bandwidth, since traffic increases along with capacity.

3.6.2 Challenges of using collaborative environments

There is always the question of why the use and popularity of collaborative environments today is not as was expected from the IT crowd. Also many analysts in the CSCW community have discussed why groupware has not always lived up to these expectations. Researchers in [74, 75, 76] identify cultural and structural problems integrating groupware into work practices. One of the problems and barriers that face the adoption of collaborative environments according to [74] is the complexity of existing tools which discourage users without continuous support from an expert. In addition, Technologies in use today do little to adapt to the task at hand, this is problematic since collaboration is never the same from one session to the next as a result of the uniqueness of individuals and the broad spectrum of tasks that can be accomplished through collaboration. Other researchers such as Markus in [77 , 78] points to the lack of a critical mass of users as a central problem in groupware adoption where groupware may not enlist the "critical mass" of users required to be useful or it can fall because it never to any one individual's advantage to use it.

Also, the research done by Grudin in [79, 80] examines some of the challenges that groupware developers should be aware of which are concerned with better knowledge of user's workplace such as the disparity between who does the work and who gets the benefit and Disruption of social processes where Groupware may be resisted if it interferes with the subtle and complex social dynamics that are common to groups such as the risk of taking inappropriate advantage of anonymity, sabotaging group work, or violating privacy. Grudin also relates the failure to adopt groupware to malfunction in the development process where it is an almost impossible obstacle to analyse and evaluate groupware and product developers face obstacles in involving users that could be particularly detrimental to groupware development [81].

Different solutions to overcome the previously mentioned challenges have been proposed in [80, 81, 82, 83], such as changes in the design stage where it is recommended to design, along with the technology, processes for using it that create benefits for all group members and demonstrate the collective and indirect benefits and try to reduce the work required of non-beneficiaries if possible. Also, adding groupware features to an already successful application rather than launch a new application with a display that creates expectations of heavy. Also emphasise on user involvement in the development have been suggested and developers need sophisticated understanding of perspective user's workplaces where working with representative users whenever is possible is a standard advice for developing interactive systems. In addition to the ongoing users' training for effective technology has been recognised.

3.6.3 Delay in Collaborative environments

3.6.3.1 Types of delay that affects groupware

Delays on the internet [84] depend on the processing power of the client machines, the bandwidth of the network segments, the distance that messages must travel, the number of routers that the message goes through, and the current traffic.

In this section we will concentrate on the research and experiments carried out by Gutwin and targets groupware user's specifically in [85], to find out what kind of delay affect groupware and the effects of network delays in wide-area networks on the usability of groupware systems. The research had primarily been concerned with groupware that involve visual artefacts in a shared workspace, and with the tasks that involve close coupling amongst the group.

For continuous real-time information, messages must be considered as part of a temporal stream. Continuous streams have temporal dependencies, in that the timing and pacing of the stream has an affect on how the stream is interpreted. Telepointer positions and other information about people's movements and activities are examples of this type of data stream. Streams are therefore sensitive to two kinds of delay that can be caused by network communication: Latency and Jitter**.**

Latency is the lag between the sending and the receiving of a message, since messages cannot be delivered instantly, latency will always exist to some degree. Even in a face-to-face conversation, there is (usually unnoticeable) communication latency due to the speed of sound. In network communication, substantially larger latencies exist, caused by the transmission time of the network medium, slowdowns due to traffic, the overhead of routing messages, and by the processing time required to unpack and process messages. From the groupware user's perspective, latency means that a data stream (such as another person's telepointer motion) is late compared to when it was produced. The motion of the telepointer will look normal in other respects, however; if the user has no indicator of when the motion started, then the latency will be difficult to detect. Problems begin to occur with latency in two situations, the first happened when two streams (such as voice and telepointer motion) are supposed to be synchronised, but are transmitted with different latencies. The second happened when interaction involves taking turns. Previous research suggests that turn-taking becomes difficult to coordinate when latency is greater than about 200-300ms, depending upon how closely coupled the task is [86, 87].

The second type of delay is jitter; Jitter affects the pacing of the stream rather than its lateness. Jitter is variance in transmission time, and measures whether the amount of time between two messages at the receiving end is the same as the time between them when they were sent. E.g. if messages are sent at 10ms intervals, but the receiving interval varies from 10ms, then there is jitter in the transmission. Jitter does not exist in face-to-face transmission, because all the data in an utterance or a movement travel at exactly the same speed. In networked groupware, however, each message in a stream is encoded as an independent packet; two consecutive messages may be sent to the destination on different routes, or may encounter different overheads and traffic conditions along the way. Furthermore, a message may be lost altogether, and if the transmission protocol enforces in-order delivery, all messages behind the lost message must wait unprocessed while it is resent from the source. These factors imply two means of characterising jitter: size of delay, and percentage of messages that are delayed. From the user's perspective, jitter appears as halting or jerky movement. E.g. a moving telepointer will appear to stick when a message is delayed, and will then catch up when messages begin flowing again. Research into the delivery of streaming audio and video suggests that people are able to notice even small amounts of jitter (tens of milliseconds) and quickly become annoyed by larger amounts [88, 89]. Audio and video applications strive to reduce jitter to zero, usually by buffering the stream before playback begins. Buffering (also called smoothing) reduces jitter by increasing the overall latency; the stream starts later but plays smoothly. People are able to notice even small amounts of jitter (tens of milliseconds) in streaming audio and video; in groupware, people have difficulty predicting telepointer movement and interpreting gestures when there are gaps of more than 600ms in the telepointer stream [90].

Another delay type is loss; *Loss* is the information that is lost in transit due to buffer overflows on network equipment, routing errors, corrupted information, or poor signal strength and interference on wireless networks. To the user, a lost telepointer message appears as a jerky motion where the telepointer jumps due to a missed frame. Losses often come in bursts, which can cause large jumps, affecting the fluidity of motion, which can be distracting and can inhibit gesture recognition.

3.6.3.2 Effects of network delay on closely-coupled social dynamics

Before starting to pinpoint the likely effects of network delays on group dynamics, I would like to clarify the meaning of “coupling”, a frequent term used in groupware. Coupling is used to refer to the sharing of information between users, including which objects get shared and how often changes get transmitted between users. Tight coupling refers to situations where almost all information is shared between users and changes are transmitted frequently (or immediately). Tight coupling can also refer to situations in which the work of one person depends on the actions of another person. If the group task requires close coupling and the network delays are high, the developers may have to rethink their design, since unreliable information about others is often worse than no information at all [91].

As we mentioned above, According to the research and experiments in [85], we concluded that even though latency and jitter did not cause major problems for the groups, the observations in [85] suggested that certain kinds of collaborative interactions were affected by delay, and deserved closer study. These involve the activities of predicting movement and coordinating access. Where in predicting another person's movement; the person's movement (moving cursor) in a shared workspace is used by others to predict where the person is going [92]. These predictions are used to anticipate actions, to join someone at their destination, and to plan motion so as to avoid bumping into one another. Furthermore, in coordinating access to shared artefacts, up to date knowledge of another person's activities in a closely-coupled situation is essential for managing access to a shared object or tool. This information is the basis of "social protocols" of concurrency control [93].

Therefore according to [85] the two most obvious effects on collaborative work were that of jitter on prediction time, and that of latency on coordination errors. These results based on experiments in [85] can be converted back to real-world terms, where with no jitter, predicting another person's pointing action took about two seconds; with jitter of 600ms, prediction took about half a second longer. With no latency, pairs made one coordination error in 50 repeated manipulations of a shared object. When latency was 240ms, they made one error in every 10 manipulations, and when latency was 1000ms, they made one error in every three.

I would like to discuss briefly the social and psychological results obtained in [85] to recognise the effects of delay in a group dynamic. In the coordination task experiment, participants were asked to drag objects from a shared central stack and drop them onto a target region. There were two drop regions, one for each participant, but only one central stack of objects. Latency in the coordination task seemed to force a "rhythmic turn-taking" strategy, where pairs used some kind of you-then-me-then-you approach to taking objects from the stack. Latency makes people unsure about what their partner was doing, and makes it difficult to determine whether it was safe to grab the next object from the stack. However, when one person appeared to be taking longer with a move, the other person would occasionally fill in the time by taking an extra object. Yet, even this strategy often broke down when latencies were larger. It was observed that it seemed clear that delays made the tasks more difficult, and they reduced satisfaction with overall systems. Jitter was less of a problem in the coordinated-moving task. This may have been because a jittery telepointer was still up-to-date most of the time, and along with the rhythm of turn-taking, may have provided enough information to keep the activity properly coordinated. One situation where jitter did appear to cause a problem, however, was when one person's cursor stuck while they were inside their drop area. This occasionally fooled their partner into thinking that they had paused, and into attempting to grab a second object from the stack.

To conclude, in the prediction and coordination tasks, a main reason that delays impaired performance is that delay introduces uncertainty into a situation where certainty is required for expertise. When visual evidence is uncertain, fast perceptual tasks (e.g. simply watching the other person) are changed into time-consuming cognitive tasks (e.g. mentally calculating the other person's location or current activity).

3.6.3.3 Dealing with delay in collaborative environments

Network delay has been shown to have serious effects on users. It can cause difficulties in coordinating collaborative actions [94, 86], in predicting others' intentions [85], and in interpreting gestural communication [95]. The overall effect is that collaboration breaks down and groups tend to decouple their collaboration and work more independently. When latency becomes extreme (as happens when systems exceed their network bandwidth), the distributed parts of the application appear to grind to a halt and telepointers freeze, locks are never granted, and changes are never propagated. However, bandwidth restrictions are one of the most critical causes of latency in distributed systems, and the size and efficiency of groupware messages will play a major role in any attempt to improve groupware performance.

There are many different genres of groupware (including games, whiteboards, conferencing systems, shared editors, virtual classrooms, and collaborative virtual environments), and each application can have multiple interaction techniques with diverse quality of service requirements. One of the most important points from the designer's perspective is to assess both the coupling requirements of the groupware system and the likely delays in the situation where the application will be installed. In addition, performance requirements are also affected by situational factors such as proximity to other users, the level of coupling and dependence between actions, and the level of awareness that a user wishes to maintain of their collaborators.

The diversity of requirements in groupware means that no one technique can solve all performance problems. Hence different methods have been suggested to deal with delay in collaborative environments.

Research into the integrity of delivering consistent collaborative environment suggests that trying to optimise the underlying network and to specify a level of performance that can be expected in the transfer of information using end-to-end Quality of Service (QoS) approaches could have beneficial influence on the usability and performance of collaborative environments [95, 96, 97, 98]. One method of such approaches provides differentiated service [99] in the sense that some classes of traffic are treated preferentially relative to other classes. Packets are marked at the edge of the network to indicate the type of treatment that they are to receive in the routers inside the network. This approach does not provide strict QoS guarantees. A second approach provides guaranteed service [100] that gives a strict bound on the end-to-end delay experienced by all packets that belong to a specific flow. This approach requires making resource reservations in the routers along the route followed by the given packet flow. Weighted fair queuing combined with traffic regulators are needed in the routers to provide this type of service [99, 100]. Groupware can be designed to auto-adopt in accordance with the negotiated level of quality of service to guarantee consistency and soft synchronization of the collaborative environment with the least rendering degradation possible. Hence, when considering user's needs, groupware systems could be designed to present the users with interaction techniques that are most appropriate for the current delay conditions. For example, as jitter increases beyond a certain level, the system could offer to switch from the use of telepointers for user awareness to a participant list, which is not affected by jitter.

A further method to deal with delay is hiding latency so the participants in a collaborative environment will not notice the delay in an interactive game, These techniques are called 'latency hiding' [101, 102] that help create the illusion of the latency being lower than it really is, but these techniques don't provide true interactivity; they only provide the illusion of interactivity. The illusion of interactivity may be better than no interactivity at all, but it is not as good as the real thing. For example, for tightly coupled interaction like sword fighting game to work, the task must be communicated over the network before you can see it and respond. A variety of latency hiding techniques has been investigated at the hardware level for example such as in [103], however, except multithreading which may require substantial program structuring effort, other software-based latency hiding methods have not been investigated, hence, new techniques for software based communication latency hiding are presented in [102]. One of the latency-hiding techniques would be locally computer prediction of reactions without waiting for the real response to come back over the network. In this case you are not really interacting with your opponent; you are interacting with your computer, but you have the illusion that you are interacting with your opponent. Buffering incoming messages is one way of reducing jitter at the cost of increasing latency.

Another method to deal with delay on a psychological level such as Revealing delay, where in [104] an approach of dealing with delay using 'decorators' is proposed. Decorators show the presence, magnitude and effects of delay so that participants can better understand its consequences and adopt their natural coping strategies. Therefore revealing delays and using explicit indications of delay, is one way in which groupware can benefit from accepting and working with the reality of distributed systems, rather than trying to maintain the illusion of copresent.

3.7 Quality of experience (QoE)

3.7.1 Quality of Service (QoS) and Quality of experience (QoE) in collaborative environments

The goal of a collaborative environment (CE) is to bring together the required people and the right data in order to perform a task, solve a problem, or simply discuss something of a common interest. At any time, one specific system or mode of interaction may be more appropriate than another. A truly successful collaborative environment should support the most appropriate tool for the task at hand, and provide other tools that can enhance collaboration.

Quality of Service of a network refers to the properties of the network that directly contribute to the degree of satisfaction that users perceive. QoS in collaborative environments is mostly concerned with the network layer and the application layer of the network protocol stack.

The network layer is concerned with the network services performed by devices such as routers and switches. In the network layer two types of QoS are available: prioritisation and resource reservation. The QoS solution for network layer includes activities such as marking, classification, admission control and scheduling. The network QoS can be applied to individual flows or aggregate. The Internet network QoS is formed by different mechanisms and solutions such as integrated services [100], Differentiated services [99], Multi-protocol label switching (MPLS) [105], and constraint based routing [106] and traffic engineering [107]. The network layer is concerned with parameters such as jitter, delay, burstiness, latency and packet loss.

QoS in data networks has really evolved to focus on packet delivery statistics. These measures can often correlate with end user system performance and from them one can draw inferences about the user’s experience. However, focusing only on mere statistics is one step removed from the actual traffic carried across the network. The ability to provide tools to users and hope that they use them effectively is not enough and providing network statistics in the form of QoS is incomplete because a user does not directly interact with, nor perceives, any of these statistics in getting some task done.

The application layer is concerned with the services provided by the application to achieve the required QoS. In the application layer the QoS is driven by the human perception of the collaborative session. For example, application metrics are, availability and continuity of service this describes the requirement for uninterrupted service with acceptable quality. There are several factors that may disrupt the continuity of the service and, among them throughput which this is the effective share of bandwidth that the application is getting from the network; in addition to Information (or data) loss. At the user-perception level, information loss does not necessarily coincide with data packet loss at the network level (mainly due to packet loss). It might be data loss at the application level (for example loss of signal fidelity due to encoding), or it can be completely user-specific. In this case, it is translated as a 'user perceivable' loss of information that leads to user discomfort, lack of timely judgement and reaction needed to execute the task successfully, annoyance, disorientation, lack of interest, etc. Therefore, information loss corresponds to the amount of information (visual content as user sees it, loss of audio clarity, etc.) that a user perceives as missing. We also have to note that this metric may have a subjective component as well. For example, a user might believe that something is wrong with a stream even in a lossless network environment. Also, more commonly, despite the fact that data are lost on the network, a user might not understand or perceive it in that way (for example, if packet loss affected a non-audible, non-visible, or non-perceivable part of the transmitted information).

It is very important to emphasize here that the above application QoS metrics are not only affected by the network-centric metrics. Several other factors in the end-to-end application path may result in undesired changes of performance parameters, like the operating system's inability to support the application, erroneous application and protocol stacks implementations, the usage environment (e.g., faulty equipment), etc.

With increasing usage of Internet services, the topic of providing adequate Quality of Service (QoS) for the Internet has become a focus of research. Traditional QoS metrics such as response time and delay no longer suffice to fully describe quality of service as perceived by users. The success of any scheme that attempts to deliver desirable levels of QoS for the future Internet must be based, not only on the progress of technology, but on users' requirements.

In addition to QoS network and application layer, according to [108] it is possible to define a perceptual pseudo-layer, which is concerned with the end experience and the metric of this perceptual layer is QoE which is defined as "the user perceived experience of what is being presented by the application layer, where the application layer acts as a user interface front end that presents the overall result of the individual Quality of Services".

Quality of experience is a concept comprising all elements of a user's perception of the network and performance relative to expectation. The concept applies to any kind of network interaction. QoE [109, 110] can also be defined as the characteristics of sensations, perceptions and opinions of people as they interact with their environments. These characteristics can be pleasing and enjoyable or displeasing and frustrating. In the current context, QoE is how the user feels about how the application or service was delivered, relative to their expectations and requirements. QoE can mean different things to different applications e.g. a high QoE for an audio application might be related to the sound fidelity and ability to smoothly take turns in a conversation. A remote video application might have a high QoE if the video image is large and clear when presented to the user. It is possible to have excellent QoS and poor QoE. Flawless transmission of garbled packets does not make for happy users. If QoE is high the user is happy and satisfied and low QoE indicates that the user does not have a good experience of the network.

QoE is receiving increased interest in the research community. For example, in [109] Bauer and Patrick have proposed a human factors extension to the seven layer OSI model which offers a common conceptual language to facilitate meaningful discussions between the Human Computer Interaction (HCI) disciplines and those responsible for network and application design. An HCI layer represents the experience that people have with the devices and services that technology offers.

In the context of collaborative environments, we find it essential to consider QoS and QoE solutions in parallel, because in the distributed collaborative environments the delay from the network can cause communication breakdown thus the QoE will fall apart. Observations in [85] suggested that certain kinds of collaborative interactions were affected by delay, and deserved closer study; these involve the activities of predicting movement and coordinating access. These predictions are used to anticipate actions and to plan motion so as to avoid bumping into one another. Up to date knowledge of another person's activities in a closely-coupled situation is essential for managing access to a shared object or tool. In the prediction and coordination tasks, a main reason that delays impaired performance as mentioned before is that delay introduces uncertainty into a situation where certainty is required.

3.7.2 Quality of Experience in VNC as a Collaborative Environment

Since VNC is a typical example of a thin client solution that can be used in distributed collaborative environments, it is interesting to investigate the QoE in VNC-based systems. I will do so by examining the use of VNC in the context of Computer Supported Cooperative Work.

Looking back at the shared workspace properties, VNC can be considered a shared workspace because it provides a desktop WYSIWIS environment where all participants view the same computer display, and can be used in research groups to facilitate their communication. Moreover, VNC is not enough for a full suite hence another channel is preferred to be used in complement to VNC such as videoconferencing or the phone.

VNC can be used as a part of a groupware, because it supports communication where the users of VNC can communicate synchronously by viewing the same desktop and interacting, and also communicate asynchronously by seeing the results of any previous session. VNC also supports collaboration in the sense that multiple researchers from all around the world can work on the same desktop and concentrate on the development of a specific application or a prototype related to their area of discipline. Moreover, VNC can be used to facilitate interaction between different parties asynchronously by using an additional tool that records the previous sessions and replay it. What VNC lacks is any form of floor control in a multi-user environment.

In order to provide the collaborative environment using VNC with a better QoE, different aspects can be added to the VNC system to enhance the collaborative experience.

The first aspect is Floor control policies that regulate access to the shared objects to preserve social rules and protocols that govern the access to the shared space. There are four floor control approaches: No control, implicit locking, explicit locking, and chair control.

For VNC, the No control approach mentioned previously is used which gives every participant the freedom to access the shared workspace and depends on common sense and rules of social behaviour to resolve any conflict. However, it would preferable to integrate a floor control tool for a multi-user environment and in turn this mainly depends on which floor control policy is chosen to be adopted. For example, in a chair control scenario a moderator is elected, to be responsible of handing over the session to who requests it. Where the moderator has a tool that lists all the clients connected referred to as numbers and IP addresses and he can hand in the control to the number corresponding to the client and cease its control too and a tool also must be developed to recognise all the pending floor requests.

The second aspect to consider is session control policy where many factors would be considered such as maximum numbers of participants joining the session, and the criteria on which a participant is permitted to join the session and avoid unauthorised intrusive access to the session. For VNC, the session control is accomplished by using a password and IP authorisation where only certain IP addresses are permitted to use the session.

The third aspect is the security of critical information; critical information must be secure even against aggressive attempts to obtain the information. While Anonymity can protect an individual there are quite legitimate reasons for identifying people for accountability, especially where security and the risk for abusive behaviour are involved. Awareness tools need to be developed and designed for letting all participants know who exactly is using the desktop. In VNC, a screen can be used to show all the IP addresses of any participants joining the session, or a program called VNCWHO [111] is used, in addition to a pop-up box asking for permission to join the session. However, who ever has access to the VNC server have full control of the machine except if he was in just a view mode, therefore he can access critical information if they are available on the server unless if access control software are enforced.

Also, asynchronous collaboration can be achieved by providing VNC system with suitable coordination mechanism; geographically separated users will be able to share workspaces and applications in a work session. And by recording the messages flowing between the client and the server, temporally separated users are capable of retrieving and playing back previous work sessions to share knowledge [112]. The recorded medium is a stream of rectangles containing pixel data of the user interface and it captures computer-based activities with no need for extra devices such as cameras or video capture cards [113], [114], [115].

The framework developed by Sheng Feng Li and Andy Hopper in [116] to integrate synchronous and asynchronous collaboration is a scalable infrastructure, which can be plugged into the client/server architecture to convert them to multi-client/multi-server for the purpose of collaboration. This framework extends the basic VNC system by creating couple of new applications, for the synchronous mode the RFB viewer had been developed while for the asynchronous mode the RFB Reviewer had been developed where it retrieves the messages stored by the framework to playback the recorded session. The recorded messages are stored in a log file and the Reviewer can then open this log file to play back the collaborative session. The messages processing performed in the Reviewer is like the messages processing performed by the viewer with the exception of that the Reviewer reads the messages from the log file while the viewer reads from the socket. This framework accomplishes that by placing a proxy server between the client/server architecture to intercept, record, redirect, merge and multi-cast the message streams between the VNC client and the VNC server.

This framework carries out some important concepts of CSCW for it to suit the prerequisite of successful coordination, this extended system can support various floor control polices. Floor control allows users to share the work session without access conflicts. A client can operate the shared work session while the rest of the collaborative clients can only view the session. And subsequently release the floor for their corresponding clients. What must be noted is this framework should only be regarded as tools to assist IT-related training where students learn about computer-based activities and these tools can be integrated into a distance-learning environment and cannot be considered enough alone as a complete suite for distance learning [115]. The developed tools are: recording and replaying, editing and annotating, browsing and searching and operating and sharing [113, 115].

Another example of using VNC in a collaborative setting is in distance learning. Distance learning facilitates education for people who cannot attend college or university for disability reasons or social circumstances. Many distance education environments that have been developed involve in a way or another three key components: a web based course material browsing system for the students, a course material editing system, and an information analysis system for the instructors [117].

In summary, when adopting VNC for remote collaboration some interaction problems will rise such as the collision of using resources and not knowing whom are you sharing the desktop with. In the case of a team sharing the desktop for research purposes or to revise a paper or to test a prototype an additional channel can be introduced to overcome interaction problems such as chat software or telephone where every user can contribute to what is happening.

There are problems in the feasibility of multi-user VNC environment which affect the users' Quality of Experience such as the ID of the author of any modification cannot be recognised or known. Hence, there should be an awareness mechanism to inform users of the author of the actions. Also, clients lost synchronisation with the server due to the overload of events processed on the server from several modifications from multiple participants because of the no feedback mechanism of who done what and the confusion of the owner of the modification.

Therefore, we conclude that in order to provide a better QoE in VNC as a collaborative environment we need:

- Floor control policies
- Session control policies
- Enhanced security
- The ability to have an asynchronous collaboration

4 Chapter Four

Overview of Thin Client Computing and Remote Control Applications

Due to the fact that VNC is considered a thin client as well as a remote control application, this section is divided into three sub-sections. The first section describes the implementation of VNC on both Windows and UNIX platforms. The second section is specifically dedicated to thin client computing and describes four thin client forms to conclude a comparative analysis with VNC. Whereas the third section looks at four popular commercial applications for remote control purposes and then establishes a comparative analysis with VNC and derives strength and weakness points and the missing features of VNC.

4.1 Implementing and Testing VNC

VNC as mentioned before is a remote display system which enables the user to view a computing 'desktop' environment from anywhere in the world through the Internet and from different kinds of computer platforms.

What sets VNC in its own league is

1. Its platform-independent feature where a PC user can view UNIX and vice versa and the user will be able to view many other architectures. Other interesting property is using web browser to view any PC or workstation without the need to install any software beforehand.
2. Its free, the user can download it from [58]; and can also distribute it and use it under the terms of GNU Public License. Both binaries and source code are available at this site as well and users can add new functions to it.

3. Its small and simple, the server is easy to install and use. Whereas the viewer of Microsoft Windows (Win32) is about 150K in size and can be saved on a floppy disk.
4. Its thin architecture, where no state is stored at the viewer. All the work is stored on the server so if the user's machine viewing the server crashes or restarts everything is left the same on the server and the user can connect to it again and find it exactly as he/she left it.
5. Its CSCW feature where many viewers can view one desktop.

The following drawbacks of X Window system that restricts using Teleporting and encouraged the development of VNC are taken from [56]

1) X requires the display machine to run an X server program. This heavyweight piece of software requires substantial resources, which machines such as NCs and personal digital assistants (PDAs) cannot be expected to run.

2) The X security model makes it inherently dangerous to allow a remote machine to use your display. Accordingly, most systems administrators stop X traffic from passing in or out of their sites.

3) Application startup is extremely slow on high-latency links due to the number of round-trips performed by a typical application (though there are special proxies that alleviate this problem).

4.1.1 Components of VNC

VNC consists of three main components

1) The VNC server, which generates a display and must be installed on the machine that the user would like to control. VNC server listens for connections on port 59xx and 58xx for the Java version, where xx is the display number between 0-99.
2) The VNC viewer, which draws the server's screen on the screen. This piece of software is small is size and easy to use.
3) The VNC protocol, also known as the RFB protocol that connects between the server and the viewer and was explained in detail in chapter 2.

VNC is available for Microsoft Windows, Linux, Solaris, Macintosh and handheld devices.

Two installation's procedures are outlined here. The first is the Windows server, which is called **WinVNC**, this procedure is a straight forward set up. While the second installation procedure is the UNIX VNC server called **Xvnc** on Solaris 2.6 and this procedure is a longer task. This thesis concentrated on Windows and UNIX VNC implementations due to their popularity among normal users and professional and its availability in our university's laboratory. Whereas Macintosh and PDA's were not included however in chapter 6.2 important sites are included for people who are interested in different platforms.

4.1.2 Installation of the VNC server and using the VNC viewer on Microsoft Windows

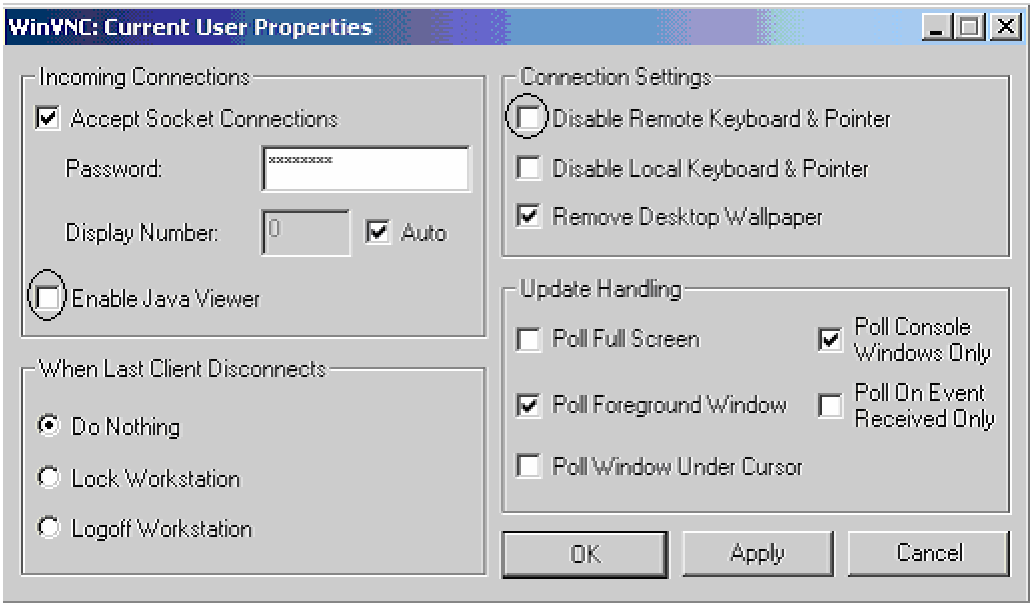
4.1.2.1 Windows VNC server (WinVNC)

A. Installing and running WinVNC

On the download screen on the VNC site, the user should download the latest version in the form of a zip file (**.zip**) and then should extract it and two folders will be found: **vncviewer** and **winvnc**. (The thesis is bases on VNC version 3.3.7)

1) Install the Windows server (WinVNC) by running the setup program included in the distribution in winvnc folder and called **setup.exe**.
2) Install the default registry settings using the option in VNC group in the startup menu.
3) Run the WinVNC server:

   Click on Start->Programs->RealVNC->Run VNC Server



Incoming Connections

☑ Accept Socket Connections

Password: *********

Display Number: 0 ☑ Auto

□ Enable Java Viewer

When Last Client Disconnects

● Do Nothing

○ Lock Workstation

○ Logoff Workstation

Connection Settings

□ Disable Remote Keyboard & Pointer

□ Disable Local Keyboard & Pointer

☑ Remove Desktop Wallpaper

Update Handling

□ Poll Full Screen

☑ Poll Console Windows Only

☑ Poll Foreground Window

□ Poll On Event Received Only

□ Poll Window Under Cursor

Figure 4.1 WinVNC user properties

and set the password. The user has the choice to uncheck the **Auto** box and choose a display number between 0-99. Otherwise whenever the viewer tries to connect to the server it causes WinVNC to select the first available display number automatically.

In Figure 4.1 another option is to check **Enable Java viewer** to allow users around the world to connect through their web browser. Furthermore, if the server's owner does not want any viewers to interfere or move anything on his machine then he should check **Disable Remote Keyboard & Pointer**.

The user knows that the VNC server is working from the system tray in Figure 4.2:

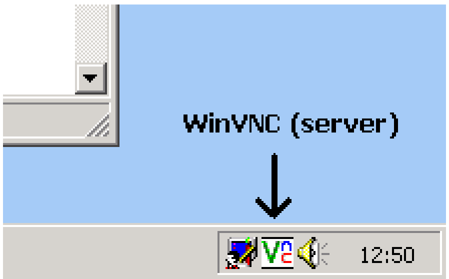


Figure 4.2 WinVNC in the taskbar

B. WinVNC Modes of operation

WinVNC can be run as:

1) An application mode: in

2) Figure 4.3 the user runs VNC whenever he needs after starting his operating system so he has the choice to run it or not depending on whether he needs it in his current PC usage session.

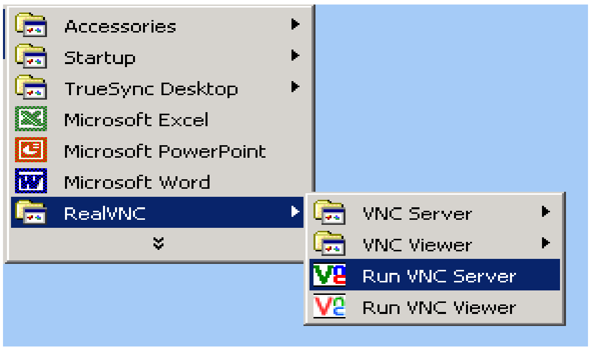


Figure 4.3 WinVNC as an application

3) As a service: in Figure 4.4 the user can connect to a machine that has nobody logged on it. The server will start up whenever the system boots up into Windows. If this is the option you are looking for then you Select "Register VNC Server Service" from the VNC Server section of the Start menu.

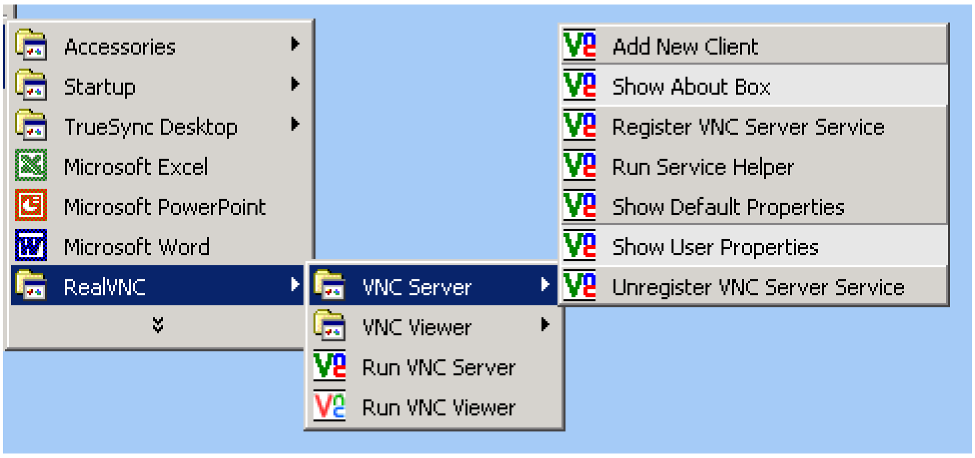


Figure 4.4 VNC server's options

In Windows 95/98/ME the server will be running immediately and will run every time the system is turned on, However in Windows NT/2000/XP the user needs to restart the system for the server to start working as a service.

C. Reverse connection

In Figure 4.4 one of the options in the VNC server is **Add New Client** which allows outgoing connections to be made from the server to any "listening" viewer, the listening viewers will be explained in the next point, where the user should provide the IP address of the desktop the VNC viewer is listening on. Connections created this way are treated as shared.

The server connects to the listening VNC viewer on the default port 5500. There is another way of initiating the connection using the **–connect** option through the command prompt which is equivalent to **Add New Client**

```
C:\ cd "\Program Files\RealVNC\WinVNC"
```

```
C:\Program Files\RealVNC\WinVNC>winvnc –connect host
```

Where

Host: is the IP address of the listening vnc viewer's machine and it connects to the default port 5500.

You can connect to a different port number if you used the option –connect as follows

```
C:\Program Files\RealVNC\WinVNC>winvnc –connect host: port#
```

Where

port# : is the port number of the listening vnc viewer

There are many other options explained in depth in [58].

4.1.2.2 VNC viewer for Windows

A. VNC viewer

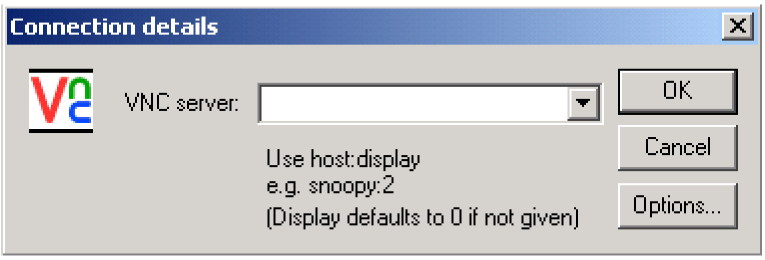
Now the only thing the user on the other part of the world needs to know to access this server is to know the IP address of the machine the server is running on and the password and if the **Auto** checkbox is unchecked the display number where the default number is 0, and it can be a number between 0 and 99. IP address is a sequence of numbers that identifies your computer on the Internet.

To find the IP address on Windows 95/98:

1. Click **Start** and then choose **Run**
2. Type **winipcfg** and an IP configuration box will pop up and it will consist of four numbers separated by periods. Each number can be zero to 255. In this form xxx.xxx.xxx.xxx

To find the IP address on Windows NT/2000/XP:

3. Click **Start** and then choose **Run**
4. Type **ipconfig** and an IP configuration box will pop up and it will consist of four numbers separated by periods.



Use host:display
e.g. snoopy:2
(Display defaults to 0 if not given)

Figure 4.5 VNC viewer

So now you should type the IP address in the box in Figure 4.5 for example:

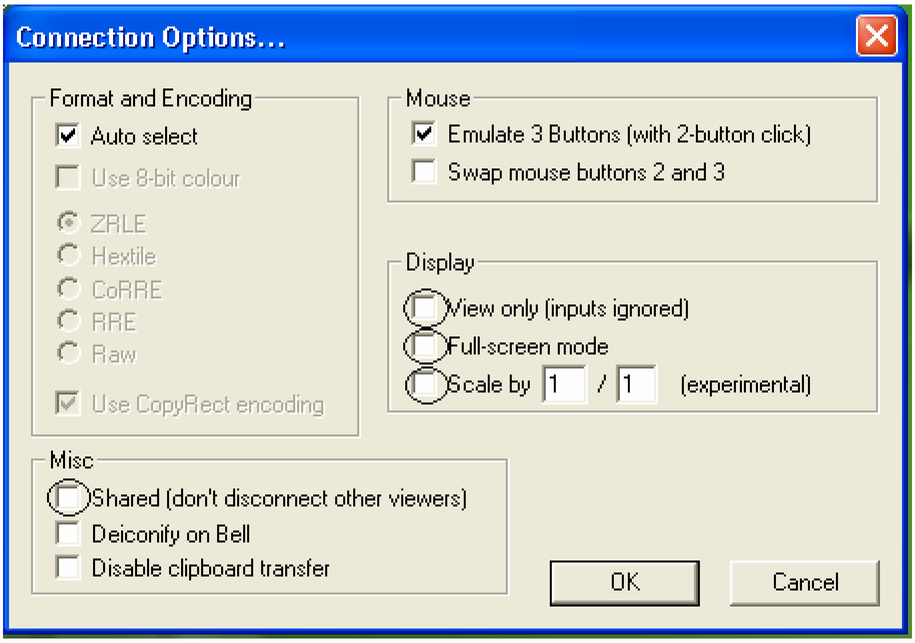
: 138.45.78.90

and without any display number if the Auto option in the server was checked. Otherwise you should type:

: 138.45.78.90: displaynumber

Where display number: a number ranges from 0-99 and must be provided from the server owner.

B. VNC viewer options



Connection Options...

Format and Encoding

☑ Auto select

□ Use 8-bit colour

● ZRLE

○ Hextile

○ CoRRE

○ RRE

○ Raw

☑ Use CopyRect encoding

Mouse

☑ Emulate 3 Buttons (with 2-button click)

□ Swap mouse buttons 2 and 3

Display

□ View only (inputs ignored)

□ Full-screen mode

□ Scale by 1 / 1 (experimental)

Misc

□ Shared (don't disconnect other viewers)

□ Deiconify on Bell

□ Disable clipboard transfer

Figure 4.6 VNC viewer's options

In Figure 4.6 the most interesting options in the viewer are:

a- When you make a connection to a VNC server, all other existing connections are normally closed; this is the default option. Therefore when the user wants to connect to a server without disconnecting any current viewers: the **Shared** checkbox must be checked.

b- If the user would like to use the viewer in a listen/view mode where he/she can only view the server's screen without interfering and moving anything, then the **View only** check box must be checked.

c- To view the server's screen in a full screen mode the **Full-screen mode** must be checked. To leave the fullscreen mode you should press Ctrl-Esc and Esc, and then right-click on the vncviewer icon.

d- To scale the screen to a smaller or larger size you should enter the ratio by which you would like to use and tick the **Scale by** checkbox.

C. Reverse Connection

This switch puts vncviewer into listening mode where it can accept connections from a VNC server. A listening vncviewer installs itself in the system tray. It can be opened from the VNC menu as well.

```
C:\ cd "\Program Files\RealVNC\WinVNC"
```

```
C:\Program Files\RealVNC\WinVNC>vncviewer –listen
```

If you would like to override the default port number 5500 you can use the *port* parameter:

```
C:\Program Files\RealVNC\WinVNC>vncviewer –listen port
```

4.1.2.3 Windows Authentication [118]

VNC authentication on Windows is a popular topic of confusion amongst users. This section will mention what is explained in [118] to understand the authentication process on Windows. The difference between the "Default" and "User" settings is not been understood well. The default settings should possibly be referred to as global settings, these are the settings inherited by VNC whenever it is started whether as a service or an application. However, VNC also allows for the assignment of unique

VNC passwords for each Windows user on a machine. Logging in as a Windows user sets these passwords, and setting the password in the user settings dialog box.

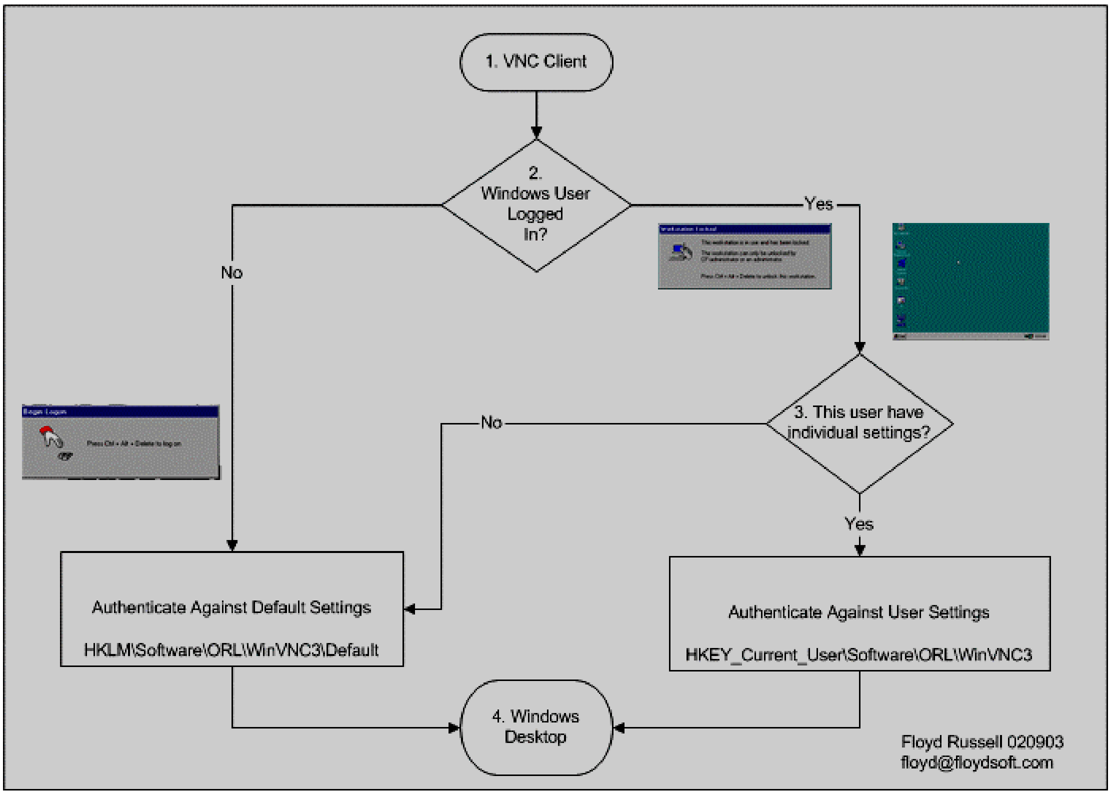

1. VNC Client
2. Windows User Logged In?
Yes
No
3. This user have individual settings?
No
Yes
Authenticate Against Default Settings
HKLM\Software\ORL\WinVNC3\Default
Authenticate Against User Settings
HKEY_Current_User\Software\ORL\WinVNC3
4. Windows Desktop
Floyd Russell 020903
floyd@floydsoft.com

Figure 4.7 VNC Windows authentication [118]

Looking at Figure 4.7

1. A user attempts to connect via VNC to a machine.

2. Is someone logged into the machine at either console or another remote connection?

3. If someone is logged in, then that user's settings are going to be used by the VNC service/application to authenticate the connecting client.

4. If no one is logged in, then the default or global settings will be used to authenticate the connecting client .

Of course if you do not want per user passwords then simply delete the registry key: *HKEY_Current_User\Software\ORL\WinVNC3*
It should be noted that you are not required to specify passwords for each windows user. In this case you must always use the Show Default Settings dialog to modify the password.

If the user wants to set passwords for every Windows user then he should follow Generic Floyd Russell's Win2k VNC configuration procedure [118]:

1. Install VNC
2. Install VNC Service
3. Start->Programs->VNC->Run WinVNC (AppMode)
4. Enter your password; this is for the current user.
5. Start->Programs->VNC->Administrative Tools->Show Default Settings
6. Enter your password; this is for the local machine
7. Close WinVNC
8. Start WinVNC Service

4.1.3 Installation of UNIX VNC server and using VNC viewer for UNIX

4.1.3.1 UNIX VNC server (Xvnc)

A. Installing the server

Xvnc is the X VNC (Virtual Network Computing) server. It is based on a standard X server, but it has a "virtual" screen rather than a physical one. X applications display themselves on it as if it was a normal X display, but they can only be accessed via a VNC viewer.

So Xvnc is really two servers in one. To the applications it is an X server, and to the remote VNC users it is a VNC server.

A single UNIX machine can run a number of VNC servers for different users, each representing a distinct VNC desktop. Each desktop is like a virtual X display, with a root window on which several X applications can appear.

The best way of starting **Xvnc** is via the **vncserver** script. **vncserver** is a perl script which simplifies the process of starting an Xvnc server. It runs Xvnc with appropriate options and starts some X applications to be displayed in the VNC desktop.

On the download screen on the VNC site, the user should download the latest version and unzip it. The following will be found:

- vncviewer: the client program
- vncserver: is a perl script to facilitate starting the Xvnc server.
- Xvnc: this is the X VNC server which is an X server and a VNC server.
- vncpasswd: vncserver script used this program to set up the password to access your X VNC desktop. It is also used to change the password.
- vncconnect: this program tells a running instance of Xvnc to connect to a listening VNC viewer.
- Classes: it's the folder where the Java VNC viewer classes are found.

1. All of these programs should be copied to a directory in the PATH environment variable such as **usr/local/bin:**

```
% cp vncviewer vncserver Xvnc vncpasswd vncconnect /usr/local/bin
```

2. If you want to enable the server to be viewed from a web browser, you should copy all the files from the classes folder as follow:

```
% mkdir –p /usr/local/vnc/classes

% cp classes/* /usr/local/vnc/classes
```

3. Now the user can use **vncserver** to run **Xvnc.** you might need to edit some lines in the **vncserver** script. For example:
   - The location of perl in the first line of **vncserver**. If it is not installed in /usr/bin then you need to edit it to point to the correct location.
   - Xvnc's font path and colour database.
   - $vncClasses: if you copied the Java classes in a different location than /usr/local/vnc/classes

A good advantage of Xvnc is that a normal user can download it that does not have root permission and unpack it in a certain directory and then to run the server is to:

```
 % cd [vnc directory]
```

and then run it.

B. Running the server:

If the user is trying to run the vncserver while he/she is geographically distant. He/she could telnet to the workstation and starts the vncserver as the following:

To run the server, it is advised to use the vncserver script, you should run it from the directory you installed it in and type:

vncserver

Now it will ask you to choose a password for your VNC desktop. Type the password twice. The program will quit and you will be returned to the shell.

To start a server, the general form is:

vncserver [:*display#*] [**-geometry** *widthxheight*] [**-depth** *depth*] [many other options]

so type:

```
vncserver :1 –geometry 1024x768 –depth 8 –nolisten local
```

The –nolisten local tells Xvnc not to use UNIX domain sockets.

The user can choose any display number from 1-250

Once you have entered the command the following will appear in the shell:

```
New 'X' Desktop is "unixserver:displaynumber"
```

Figure 4.8 XVNC Desktop

Any option can be chosen depending on its suitability to the user's needs. Some of the interesting options that are useful when the user is using many applications are:

-geometry *widthxheight*

Specify the size of the desktop to be created.

-depth *depth*

Specify the pixel depth in bits of the desktop to be created.

-cc 3

As an alternative to the default TrueColor visual, this allows you to run an Xvnc server with a PseudoColor visual.

-kill :*display#*

This kills a VNC desktop previously started with vncserver.

These are the Xvnc options that also can be passed to the vncserver script

-rfbport *port*

Specifies the TCP port on which Xvnc listens for connections from viewers. The default is 5900 plus the display number.

-alwaysshared

Always treat new clients as shared.

-nevershared

Never treat new clients as shared.

-dontdisconnect

Don't disconnect existing clients when a new "non-shared" connection comes in. Instead the new connection is refused. New "shared" connections are still allowed in the normal way.

-**query** *hostname*

Contact named host for indirect XDMCP.

-**once**

Terminate server after one session.

For more information refer to [58].

C. Issues with the implementation of Xvnc

- One of the first problems the user might encounter is that Xvnc or vncserver or even vncconnect can only be used as a root so only a super user can use them.

That's because Xvnc cannot create the UNIX domain socket in /tmp/.X11-unix. To solve this problem you should ensure that users can write to this directory by making it world-writable by doing the following:

```
% chmod 01777 /tmp/.X11-unix
```

- Another problem is when the viewer connects to Xvnc the user only can view a grey desktop with a cursor, the reason is the vncserver script after initiating the server will run $HOME/.vnc/xstartup script and by default will try to start twm window manager. If this window manager is not in your path or you would like to include another window manager then you should edit the script as follow:

```
% cd →to your home directory
```

For example:/export/home/dana/.vnc and xstartup will be in this directory

```
%vi xstartup
```

The following script will show:

```
1. #!/bin/sh
2. #xsetroot –solid grey
3. #xterm –geometry 80x24x10x10 –ls –title
4. "$VNCDESKTOP Desktop" &
5.  twm &
```

The fifth line must be edited to the window manager you would like to use. For this research we used dtwm so the fifth line will look like this:

5. dtwm &

So now whenever you want to start up vncserver and to display the dtlogin screen. you should type:

vncserver –query localhost

In addition to any extra options feasible to you, this option will allow only users with accounts on the UNIX workstation to connect in the first time and afterwards if the server is still running any other user who can supply the password of the VNC server can connect to the same server.

- The password must be at least six characters long, and only the first eight characters are significant. Note that the stored password is **not** encrypted securely - anyone who has access to this file can trivially find out the plaintext password, so **vncpasswd** always sets appropriate permissions (read and write only by the owner). However, when accessing a VNC desktop a challenge-response mechanism is used over the wire making it hard for anyone to crack the password simply by snooping on the network.
- One of the problems I faced while viewing a UNIX vnc session on Solaris using a windows viewer is the inaccuracy and the scrambled display of some applications such as Opera. Therefore When a user would like to view some X applications which use pseudocolor other than the default true colours the vncserver must be invoked using the following option:

  vncserver –cc 3

  This solution had worked effectively and the applications were displayed accurately on the viewer's screen.

- UNIX implementation of VNC is a multi user environment so many vnc servers can run on the workstation on different displays, this is shown in **Appendix A.** also for setting VNC on Linux and setting up separate X session for multiple users refer to [119].

D. Reverse connection

vncconnect tells an Xvnc server to make a "reverse" connection to a listening VNC viewer. The general form of vncconnect is:

```
vncconnect [-display Xdisplay] host[:port]
```

where

-display *Xdisplay*

Specifies the Xvnc server to control

Host: is the IP address of the machine that the vnc viewer is listening on.

Port: by default it is 5500 and does not need to be written, except when the port number on the viewer's machine is been changed.

So now you should type:

vncconnect –display unixserver:displaynumber 138.45.78.90

Where unixserver:displaynumber is taken from Figure 4.8

138.45.78.90: is an example of the IP address of the listening viewer's machine.

E. Shutting down the VNC server

To shut down the Xvnc server you should kill the process either by:

The –kill option in the vncserver script:

vncserver –kill :displaynumber

or by using the following steps:

```
% ps –a
```

A list of processes and process IDs are listed and Xvnc will be one of them.

```
% kill -9 processid
```

where the *processid* is the number of the process ID of Xvnc

and then the following files should be deleted

```
% cd /tmp/.X11-unix
%ls
```

This list will show

```
% X0 X1 X2                  which are the vnc displays working except X0 which is
                            normally X server hence they should be deleted
% rm X1 X2
                            rm is the remove command
```

And then:

```
% cd $HOME/.vnc             where $HOME is your home directory
% ls
% xstartup  passwd  unixserver:1.pid  unixserver:1.log unixserver:2.pid
                            unixserver:2.log
```

Where:

unixserver:1.log and unixserver:2.log are the log files for the VNC session on display number 1 and display number 2 and unixserver1/2.pid is the servers' process IDs.

```
%rm unix*
```

4.1.3.2 VNC viewer for UNIX

A. VNC viewer

vncviewer is a viewer (client) for Virtual Network Computing.

If you run the viewer with no arguments it will prompt you for a VNC server to connect to. Alternatively, specify the VNC server as an argument as follow

```
vncviewer [host][:display#]
```

where

host: is the IP address of the workstation

display#: is the display number which is shown when you run the vncserver .

One of the issue while using your UNIX vnc server is that when the user is working on his/her UNIX workstation he would like to remote the normal X display of his/her workstation. On UNIX environments the Xvnc server acts as a separate display independent of the existing screen display. The standard behaviour of VNC on UNIX platforms is to start a completely new X desktop(:n) which is independent of the standard X display of the workstation (:0); viewing the normal X display was not implemented in VNC however version 4.x of VNC has support for remoting the :0 display can be achieved.

This can be also be resolved by running a VNC server on your machine on a specific display and connecting to this server using the **vncviewer** on the same machine by typing

```
% vncviewer [host][:display#]
```

Where

host: is the IP address of the workstation

display#: is the display number which is shown when you run the vncserver.

```
% vncviewer unixserver:displaynumber
```

where unixserver:displaynumber is taken from Figure 4.8. After that the user should do all the work on the local vncviewer hence when he wants to view it later or would like to ask someone to share viewing it he just needs to connect to that specific VNC session again from the same machine or using another remote machine and he will view exactly the same session as he last left it.

The VNC viewer can have many options such as:

-shared

When you make a connection to a VNC server, all other existing connections are normally closed. This option requests that they be left open, allowing you to share the desktop with someone already using it.

-viewonly

Specifies that no keyboard or mouse events should be sent to the server. Useful if you want to view a desktop without interfering

-fullscreen

Start in full-screen mode.

-listen *[display-number]*

Causes vncviewer to listen on port 5500 + *display-number* for reverse connections from a VNC server

For the rest of the options refer to [58] or the help command as

```
% vncviewer –h
```

B. Reverse connection

To command the vncviewer to act as listening daemon then you should invoke the viewer using the following command:

```
% vncviewer –listen
```

And any other options can be included depending on your requirements.

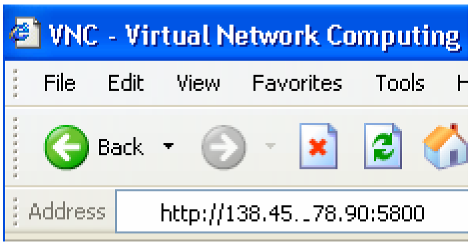
4.1.4 The Java VNC viewer

The VNC server contains a small web server so any user with a Java-capable web browser can view the desktop computing environment as an applet embedded in his/her web browser. This happens when you point to the server using your web browser, a Java version of the viewer will be downloaded on your machine and you will be asked to enter your password and after the authentication you will be able to have the remote desktop screen in front of you in the web browser.

The server listens for HTTP connections on port 5800 plus the display number. For example if the display number on a specific machine was 55 then the port will be 5800 + 55= 5855 and so on.

Note that this is the HTTP port used for downloading pages and applets, but once the applet is running it uses 59xx for VNC just like any other viewer.

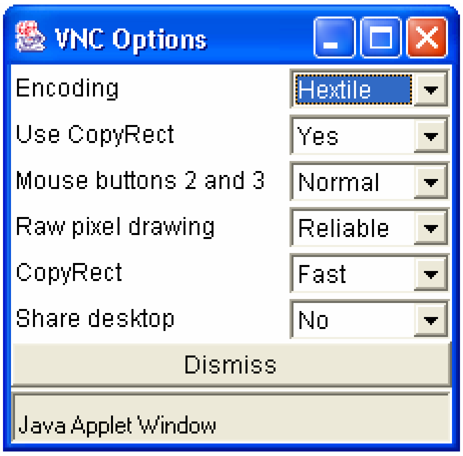
So to view display 0 on machine which its IP address is: 138.45.78.90 refer to Figure 4.9



Address http://138.45._78.90:5800

Figure 4.9 Connect to VNC server via browser

An authentication screen will appear when you connect where you will be asked to enter your legitimate password. The screen has different options, it is worth experimenting with the Options shown in Figure 4.10 especially the **Encoding** and **Raw pixel drawing** to test which is the faster method depending on your connection. **Share Desktop** is the option where you specify if you would like to share the desktop you are connecting to with any current users in the case that the server is not already set in its options to be shared.



Encoding	Hextile
Use CopyRect	Yes
Mouse buttons 2 and 3	Normal
Raw pixel drawing	Reliable
CopyRect	Fast
Share desktop	No

Figure 4.10 VNC options

4.2 Overview of Thin Client Computing

The application logic is separated from the user interface at the server and transported to the client. This separation means that only screen updates, mouse clicks, and keystrokes travel the network to the server therefore only a fraction of the usual network bandwidth is required because fewer resources are necessary by the applications. Deciding how and when to use thin-client/server technology is matter of optimising the professionals' time and computing resources [120].

4.2.1 Essential forms of thin-client/server computing

A. Citrix's MetaFrame.
B. Graphon's Go-Global
C. Microsoft Window-based terminal server (previously known as hydra).
D. SCO's Tarantella

4.2.1.1 CITRIX's thin-client/server architecture [3][121]

Citrix consists of three components

1. MetaFrame is a server-based software product that can be used to provide users the access to applications hosted either on a Windows-based environment or UNIX-based environment.
2. ICA (Independent Computing Architecture) Client software must be installed on the client device, so users can connect to the MetaFrame server from a client device, such as a Windows PC. The ICA Client software is provided free, and is available for a range of different devices, allowing users to connect from various platforms. Because applications run on the server and not on the client computer, users can connect from any client device. A Macintosh, a Windows PC or another UNIX machine can be used. The same client is used to connect to either MetaFrame for UNIX or MetaFrame for Windows.

[3] Sitting on top of the Microsoft Windows NT Server.

3. ICA protocol to send information between the client device and the server. This protocol divides the application execution from the display logic and only sends the keystrokes, mouse clicks, audio and screens updates between the server and the client. The application processing remains on the server, which means that processing on the client is kept to a minimum.

MetaFrame for UNIX uses the security set up on the UNIX server. Therefore, the user at the client device can log on using their existing UNIX user account and password. The administrators can control which users or groups of users can use particular MetaFrame features, such as logging on, disconnecting and sending messages to other sessions, using the MetaFrame security feature. A technique called shadowing is used to display and interact with another user's session and can be useful in helping remote users with training and technical support issues.

When configuring MetaFrame many aspects can be specified such as the maximum number of ICA sessions allowed to connect to the server, What happens to a session if the connection is broken or times out, Whether to allow shadowing, The maximum permitted session duration, and how long to leave idle or disconnected sessions before timing out. The Citrix ICA Client sends a packet to port 1494 on the Citrix server requesting a response to a randomly selected port above 1023.The Citrix server responds by sending packets to the Citrix ICA client with the destination port set to the port requested by the client. Citrix supports high colour (16 bit) and true colour (24 bit).

4.2.1.2 Graphon (Go Global) [122]

Go Global consists of three components

1. **Go-Global Servers**

Where GO-Global XP is a platform for Web computing that delivers 32-bit Windows applications to virtually any computer connected to the Internet. With GO-Global installed on a standard Windows NT 4.0 or Windows 2000 Server, applications can be run from any desktop computer, including Macintosh, UNIX, Linux, network computers, and most versions of Windows. GO-Global server must have TCP/IP as a network protocol. A Web Server (e.g., Microsoft Personal Web Server, Microsoft IIS, Netscape Suitespot, Netscape Enterprise Server, Apache Web Server, etc.) must also be available in order to set up the server for browser deployment of GO-Global.

Windows or Linux desktop devices can run UNIX applications by using Go-Global UX. A UNIX/X application can run over the Internet without a local X server with GO-Global for UNIX and UNIX can be run from any Windows or Linux desktop without emulation software to bring specialized applications to otherwise incompatible display devices. As illustrated in Figure 4.11.

GO-Global provides a facility for transferring files between the PC and the remote UNIX host. File transfers can be initiated at any time from the host display. Since the transfers are done in the background, you can continue to use GO-Global during a file transfer, even on a serial connection.

Most X Window processing is accomplished in its native UNIX environment by the GlobalHost software, with drawing and screen updates performed on the desktop by the client.

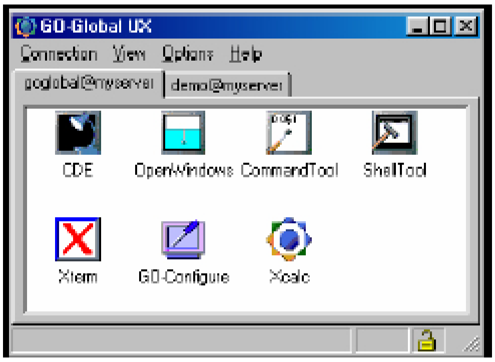


Figure 4.11 Go-Global UX [122]

2. **Thin RapidX protocol** which is used for communication between the server and the client, GraphOn's RapidX protocol makes efficient use of network bandwidth, allowing for high performance whether on a LAN or low-bandwidth lines, such as dial-up, ISDN, PPP, and wide area networks (WANs).
3. **Go-Global clients**: Java client supports Netscape Navigator and Internet Explorer, or a Netscape Plug-in, Microsoft ActiveX Control, Windows clients and Linux clients. The Thin client is less than 100KB loaded at the desktop.

Go-Global provides support for applications which require 24 bit (TrueColour) depth and addition to 8 bit colour mode. One of its features is Adaptive GUI, which is an Auto-detection of application screen depth and bandwidth. It also exploits compression for data transmission between the server and client. On low-bandwidth machines, enabling compression will increase session performance but high-bandwidth connections may actually experience reduced speed, due to processing time.

4.2.1.3 Microsoft Terminal server Edition [123], [124]

1. **Windows-based environment**

Terminal Services provides remote access to a server desktop through "thin client" software, serving as a terminal emulator. Terminal Services transmits only the user interface of the program to the client. The client then returns keyboard and mouse clicks back to be processed by the server. Each user logs on and sees only their individual session, which is managed transparently by the server operating system and is independent of any other client session. Client software can run on a number of client hardware devices, including computers and Windows-based Terminals. Other devices, such as Macintosh computers or UNIX-based workstations, can also connect to a Terminal server with additional third party software.

Previously known as Microsoft Windows NT hydra, this thin-client solution allows users to run the 32-bit windows desktop operating system and Windows-based applications completely off the server on PC and non-PC desktops. This can be achieved when users access those applications using one of the following types of desktops:

1. Windows-based terminals: low-cost hardware.
2. Windows desktop operating system either:

   Existing 32-bit operating system: Windows NT, Windows 95/98.

   Old 16-bit operating system: Windows 3.11.

   Both types should run a Window NT hydra client, within the local desktop environment.
3. X-based terminals, Macintosh, NC, MS-DOS, and UNIX based desktops.

As can be seen from the above this method lacks flexibility, where it is windows dominated and has the limitation of accessing only windows-based applications while it does not support the reverse direction of providing the ability to access non-windows applications unless a third party software is used as explained later. These limitations restrain the usage of this solution.

There are three components of the Windows 2000 terminal server edition:

1. Windows NT terminal server: a multi-user server core, which is able to host multiple, simultaneous clients sessions.
2. Remote Desktop Protocol (RDP): this protocol is responsible of the communication between the Windows 2000 terminal server and the super thin client over a network. RDP is based on International Telecommunications Union (ITU) T.120 protocol, an international standard multi-channel conferencing protocol currently used in the Microsoft NetMeeting conferencing

software product. It is capable of delivering the data on "Real-time" basis from an application to multiple clients and this property is called "multi-point delivery". The activity where the server sends or receives a data packet through the RDP stack corresponds to what happens in the seven layers of OSI (Open system Interconnection) model. RDP specification is now available under license for a one-time licensing fee, the RDP specification is designed for third parties that need to understand RDP in order to create new products that can use RDP like non-windows clients.

3. Super-thin client: this client software displays a Window user interface on a range of desktop hardware mentioned above.

The terminal server client sessions loads separate drivers for the display, keyboard and mouse:

1) (RDP) Remote Desktop Protocol and Display driver (rdpdd.dll)
2) Mouse and keyboard drivers are replaced with the RDP driver (rdpwd.sys)
3) A listener thread to detect on a TCP port any request for client connection (Termdd.sys)

The RDP client can be installed on a window-based desktop and it is approximately 70KB in size. In addition to that it can be installed on non-window-based desktop and its size is approximately 130KB. Both types use 300KB working set and 100 KB for display data. By default, connections to terminal servers are secured by 128-bit, bi-directional RC4 encryption—when used with a client that supports 128-bit.

2. UNIX-based environment

While it is possible to use Terminal Server to deploy applications for Windows to UNIX desktops and X-terminals, you may want to access existing UNIX systems using

Windows-based Terminals. Deploying a third party application, GO-Between from GraphOn Corporation [122], access to X-Windows applications can be provided from any Terminal Server thin client. GO-Between centralizes X-Windows application processing on a UNIX host. Using a proprietary protocol called RapidX, a lightweight protocol that carries only application screen updates and application user interaction, GO-Between executes X-Windows applications on the UNIX host and sends just the necessary screen updates down the wire. Deploying a small RapidX client (less than 300 KB) on the user's Terminal Server, the user has complete access to his/her UNIX X-Windows applications.

4.2.1.4 Tarantella Enterprise software [125]

Using tarantella Enterprise 3 software, users can remotely access applications running on different platforms like Windows, UNIX, Linux and more from their client machines using a web browser such as Microsoft Internet Explorer or Netscape Navigator with Java technology without the need to install any additional software on the designated client devices, which like any thin-client server architecture provides many advantages such as centralization of all the applications and management in the server which eliminates the need to install the software on the client devices.

Tarantella Enterprise 3 comprises of three components as shown in Figure 4.12:

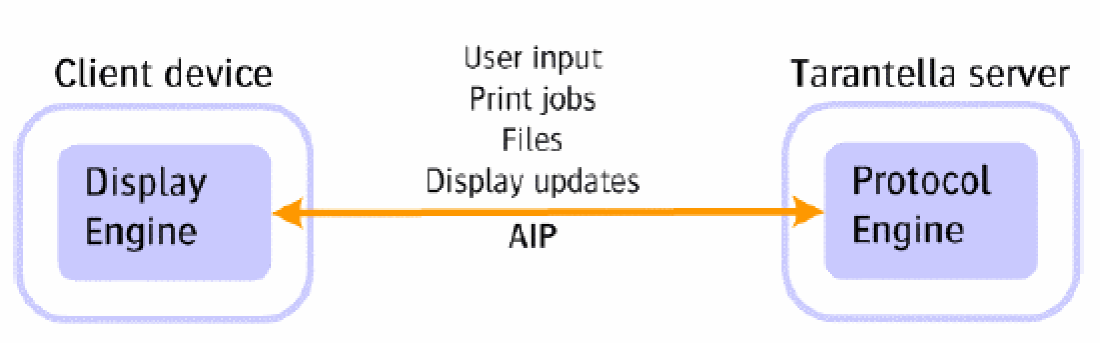


Figure 4.12 Tarantella Enterprise 3 components [125]

1. Tarantella server: which acts as a shield between the application server and the client software, the client device and applications server can connect only to the tarantella server and not to each other.
2. Adaptive Internet Protocol (AIP): AIP is used between the Protocol Engine running on the Tarantella server and the Display Engines running on the client device. AIP carries user input and display updates plus print jobs and files accessed through the client drive mapping feature. When used in conjunction with the Tarantella Security Pack, all AIP traffic is encrypted.
3. Tarantella clients: web browsers and standalone applications, called Native Clients, client enabled devices including thin clients, wireless PDAs and pocket PCs.

A Display Engine on the client performs application visualisation that receives display updates from and sends user input to a Protocol engine running on Tarantella server. Protocol engine is responsible for communicating with applications servers using appropriate native protocols such as RDP, ICA, X11 and others.

Tarantella Enterprise 3 software acts as a data store that maintains all the information about users and applications. The technique that this solution follows is when the users authenticate themselves to the server, the server:

1) Checks the user's permission level to access the applications.
2) Then it presents the application in a “webtop” form.

The software is responsible for managing connections, user sessions and security.

There are many applications to this solution; ASPs (Application Service Providers) and other service providers can benefit from it, where there target market is a large number of users with diverse client devices and diverse connectivity types.

In some cases a tarantella native client can be provided for some client devices in order to access applications using a native software rather than Java technology.

When the user points his/her web browser to the URL of the tarantella Enterprise 3 server for the first time, Java archives are downloaded and installed on the device which takes longer than it is in next accesses, these Java archives contains Java applets that are responsible of performing specific functions:

1) The framework Applet: This connects initially to the tarantella Enterprise 3 server and stores client-related state information throughout a user's session.
2) A proxy Applet: the function of this applet is to determine the identity and configuration of the server that is used by the proxy server to suitably route the traffic.

A login Applet is displayed that contains the user name and the password which the user must fill and log in to pass those details to the server where those details are compared against the username and password database and therefore identify the appearance and contents of the user "webtop", once authenticate the "webtop" applet is displayed to the user with the applications he/she is permitted to access.

4.2.2 Comparative Analysis

As mentioned before in the introduction there is two approaches to thin client computing: Web-based approaches and graphics pipeline interceptions. [1]

Citrix MetaFrame, SCO Tarantella, Graphon RapidX and AT &T VNC and Windows Terminal Services, implement the graphics pipeline interceptions. The problem with this approach is its need for a high bandwidth connection due to the fact that the virtual frame buffer is sent from the server to the client. To address this problem a lossy compression is proposed in [56] but using it might raise problems because it requires special hardware and will introduce unwanted compression artefacts into the display.

When comparing between thin client platforms, five characteristics are considered as tested in Columbia University [126], [127], [128], [129]

1) Display primitives encoding: is the type of primitive used by the remote display protocol for transmitting screen updates.
2) Encoding compression: is the type of compression that is applied to the display encoding to reduce the amount of data transferred for screen updates.
3) Update policy: is the policy for determining when screen updates are sent from the server to the client, that may be adopted based on the availability of network bandwidth.
4) Client cache: is a cache on the client that can be used to cache display primitives that are reused so that they do not need to be resent from the server.
5) Supported display colour depth and transport protocol.

The first criteria is display encoding primitives, there are four types of display encodings: first, high level encoding are more bandwidth efficient however it requires more computational complexity on the client and is less platform independent like X Windows system, second, low level graphics encoding like Citrix's ICA, Microsoft terminal services' RDP and Tarantella's AIP. Third, 2D draw primitives such as VNC. Fourth, Raw pixels: it can be enforced in VNC but is rarely used.

The second criteria and fourth criteria are compression and caching, Citrix's ICA and Microsoft terminal services' RDP employ Run Length Encoding compression and cache fonts and bitmaps in memory and on disk at the client. Tarantella's AIP employs local client caching of display objects and uses adaptive mechanism to progressively enable higher degrees of compression as the availability of network bandwidth becomes limited, VNC has Run Length Encoding compression built-in with its display encoding format and employs a very simple form of on-screen caching where by the client can simply copy display from one portion of the screen to another instead of requesting it from the server if the display data is already displayed on another portion of the frame buffer.

The third criteria is the update policy where there are two issues to discuss. One of them is whether it is a server-push or client-pull model and the second is whether it is eager or lazy display updates. In the server-push model it is the server's responsibility to determine when to send updates to the client while in the client-pull model whenever the client wants a screen update it sends a request to the server.

The second issue is if the display updates are sent eagerly with the server window system graphics commands or lazily as a framebuffer scraper. In the eager mode, the display update is encoded and sent at the time the server window system command occurs. In the lazy case intermediate screen updates are discarded and overwritten by newer updates and only the latest screen updates are encoded and sent at regular intervals depending on the available bandwidth. Citrix's ICA, Microsoft terminal services' RDP and Tarantella's AIP employ a server push mechanism while only VNC employs a client-pull mechanism. Citrix's ICA, Microsoft terminal services' RDP and VNC use a lazy display update policy which is bandwidth efficient but is incompatible with the needs of multimedia applications like video. Tarantella's AIP employs lazy or eager update model depending on the bandwidth load. Eager server-push models provide better overall performance than lazy update models like RDP, ICA, and VNC especially for multimedia applications while lazy updates can save bandwidth it degrades performance of multimedia applications even in high bandwidth environment [126].

According to [126], [127], [128], [129] ICA, RDP and VNC were able to deliver sub-second average web-pages latencies over bandwidth as low as 768 kbps (DSL). The best performing thin-client systems at the LAN bandwidths were X and AIP. Only X, AIP deliver good video quality at the highest bandwidth, none of the platforms deliver reasonable video quality at lower network bandwidths. In high bandwidth environment only VNC had high CPU load which frequently peaks to 100 percent on the client side, suggesting that VNC video performance appear to be limited by the client's CPU speed. AIP and X are able to support a broader range of applications while ICA, RDP and VNC can be quite bandwidth efficient for web applications but degrade the performance in multimedia video applications.

ICA XP , VNC, AIP and RapidX support up to 24 bits while RDP only supports 8-bits colour depth.

4.3 Overview of Remote Control Applications

Driven by changing business needs and shifting social trends, more workers are roaming further from their offices. Remote Control is a way of reaching out to these workers based on telecommunications technology.

Remote Control software are very popular amongst Helpdesk operators, network administrators, and other IT professionals and are used for troubleshooting computer problems and supply the ability to view another person’s computer screen, check and modify settings, and restart the computer without moving from their computer. Moreover, Network administrators use remote control software to connect to servers within their organization and perform routine maintenance, assess performance, and troubleshoot network issues.

Another important issue considered is to support the internal infrastructure because customer service is an essential resource for a successful IT department. Using Remote Control Software will achieve an important goal of using a Help Desk in assisting as many users without the need to be physically available at the remote site, this improves the responsiveness of the Help Desk staff and saves a lot of effort, money and time and allows the Help Desk to reach geographically distant sites without actually being there.

Of the important requirements of a Remote Control software is that the software must be Secure, Scalable and easy to deploy and maintain. Scalability is the ability of a computer application to continue to function well when it is changed in size or volume in order to meet a user need, or in other words it is the ease with which a system or component can be modified to fit the problem area. Easy deployment is where it allows pushing silent installation with wizards to guide the user through installation process.

It is worth pinpointing the difference between remote control and remote networking to avoid confusion. Remote networking let you dial into a network server using a modem and it is sometimes referred to as (dial-up networking on Windows 9x and Windows ME or remote access service (RAS) on Windows NT and 2000). Whereas remote control is the ability to view the desktop and control the whole environment and use and manipulate it's resources.

NetSupport, Carbon Copy, pcAnywhere and Unicenter remote control packages are produced for computers running Microsoft Windows Operating Systems. But NetSupport can connect to and remote control a Linux or Mac based systems that run a VNC server.

4.3.1 Remote Control Software

4.3.1.1 NetSupport [130]

NetSupport range comprise of powerful Remote Control, Enterprise Management and IT training products which comes in several languages including Spanish, German, Italian, Japanese, Brazilian and Portuguese.

In the NetSupport Manager environment they use different terms describing their Client/Server architecture: The PC that views and controls the remote PC is called

"Control" while it is in VNC terms called the "viewer". Whereas the PC that is viewed or controlled is called "Client" while in VNC terms is called the "server". These terms are confusing and do not satisfy the client/server architecture traditional terms.

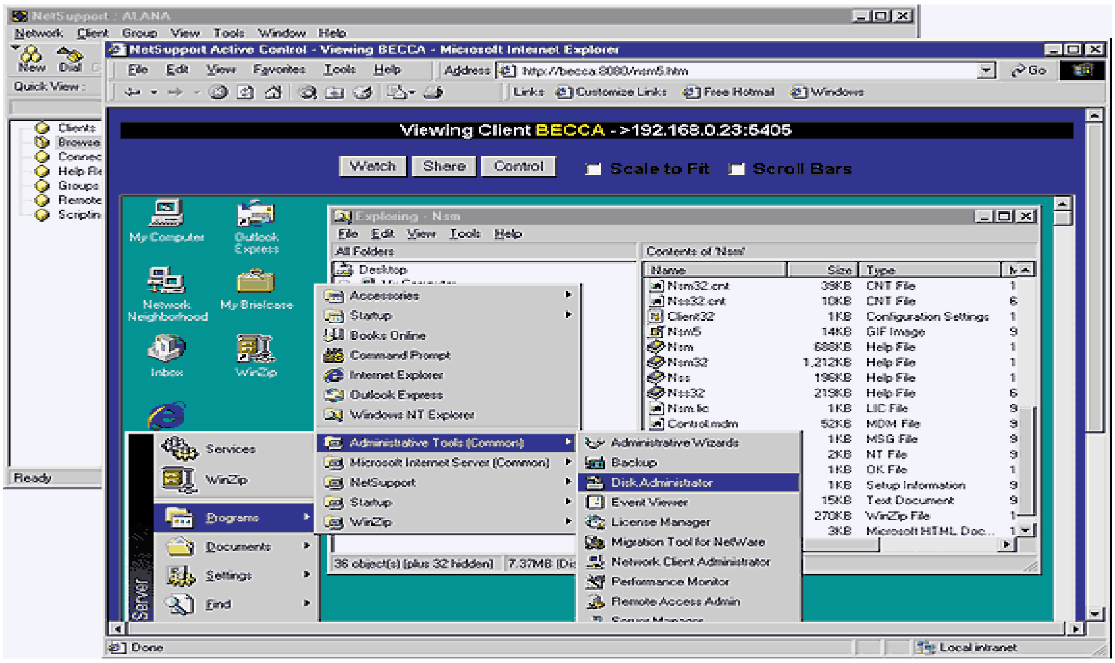
NetSupport can connect to and remote control a Linux or Mac based systems that have previously installed a VNC server. In their documentation, they confuse the user with using the term VNC client to represent the VNC server that should be installed on Mac and Linux systems in order for the NetSupport Manager to view it and control it, nothing of NetSupport software is installed on these workstations.

NetSupport Manager is a Remote Control package, which has several features, such as Scripting and Scheduling to automate tasks such as file transfer, Data retrieval and software updates, Remote Deployment to install the NetSupport client on the remote computer you can use the Deploy module to roll out NetSupport from a central site without leaving you chair, File transfer using advanced "drag & drop technology" to transfer files between different workstations and encryption can be used to ensure security while transferring files. And it also has the ability to disable file transfer or specific files and directories. It also can blank the client screen when the control is viewing its screen. And it also performs scaling to fit the size of the window you use without any scrollbars. As well as a Scan function that enables the control to cycle through each connected client for a given interval of time and displays it on the control screen. Multiple client screens can also be scanned in one scan window. NetSupport Gateway feature provides a means of connecting clients and controls across the Internet, thus delivering web based remote control without the need for modifications to existing firewall configurations. The gateway uses HTTP port 3085. There is an exhibiting mode where you can show other clients a client you are viewing.

In addition to that it provides a comprehensive security features from a password to a built-in NT security and DES/AES encryption. Moreover, it offers a message and a chat facilities and a help request to the Help Desk if users need any assistance and hence the Help Desk connects to the user's machines to provide their support; in addition to watch-only mode for the control. The software works on different types of communication like dial-up, ISDN, Internet or Direct serial link. NetSupport also supports multimedia (Audio and Video).

NetSupport Manager runs on DOS, Win3x, WFWG, Win9x, WinNT, Win ME, OS/2, Win 2000, XP(home and professional editions) and can operate on a number of network protocols simultaneously which provides an enhanced flexibility for remote sites with different topology and standards like IPX, NetBIOS, TCP/IP and HTTP . If the user is using TCP/IP the address is in the form >192.168.100.20, if using the IPX network support the address is in the form >000001-12345678, if using NetBIOS, the client's PC address is the registered NetBIOS network name of the PC.

NetSupport offers a recording facility for remote control sessions and save it in a Replay file which is useful in training sessions so if users need to ask a question about a pervious topic covered in an earlier training session he/she just needs to replay it to find the answer.



NetSupport Active Control - Viewing BECCA - Microsoft Internet Explorer

Address: http://becca:8080/nsm5.htm

Viewing Client BECCA ->192.168.0.23:5405

Contents of 'Nsm'

Name	Size	Type	M
Nsm32.cnt	39KB	CNT File	1
Nss32.cnt	10KB	CNT File	6
Client32	1KB	Configuration Settings	1
Nsm5	14KB	GIF Image	9
Nsm	688KB	Help File	1
Nsm32	1,212KB	Help File	1
Nss	196KB	Help File	1
Nss32	219KB	Help File	6
Nsm.lic	1KB	LIC File	9
Control.mdm	52KB	MDM File	9
	1KB	MSG File	9
	2KB	NT File	9
	1KB	OK File	1
	1KB	Setup Information	9
	15KB	Text Document	9
	270KB	WinZip File	1
	3KB	Microsoft HTML Doc...	1

36 object(s) (plus 32 hidden) 7.37MB

Figure 4.13 NetSupport browser control

The server can act as a simple Web server, and NetSupport provides an example browser interface. A user can remotely control a workstation from within your Internet browser such as in Figure 4.13. Using an Internet connection, the user can download the ActiveX Control software from the Website.

Users of NetSupport are AIR FORCE, IBM, NatWest Group, British Gas, BT, VOLVO, Lloyds, AOL, ORACLE, BARCLAYS and many other companies.

4.3.1.2 Altiris Carbon Copy [131]

Carbon Copy is a Remote Control software used to improve remote access and enhance help desks, it connects to remote users over network, Internet, dial-up or Direct connections including support for Virtual Private Networks and firewalls, it copies files from and to the remote node. In addition to that it provides voice chat to communicate. The carbon copy interface is shown in Figure 4.14.

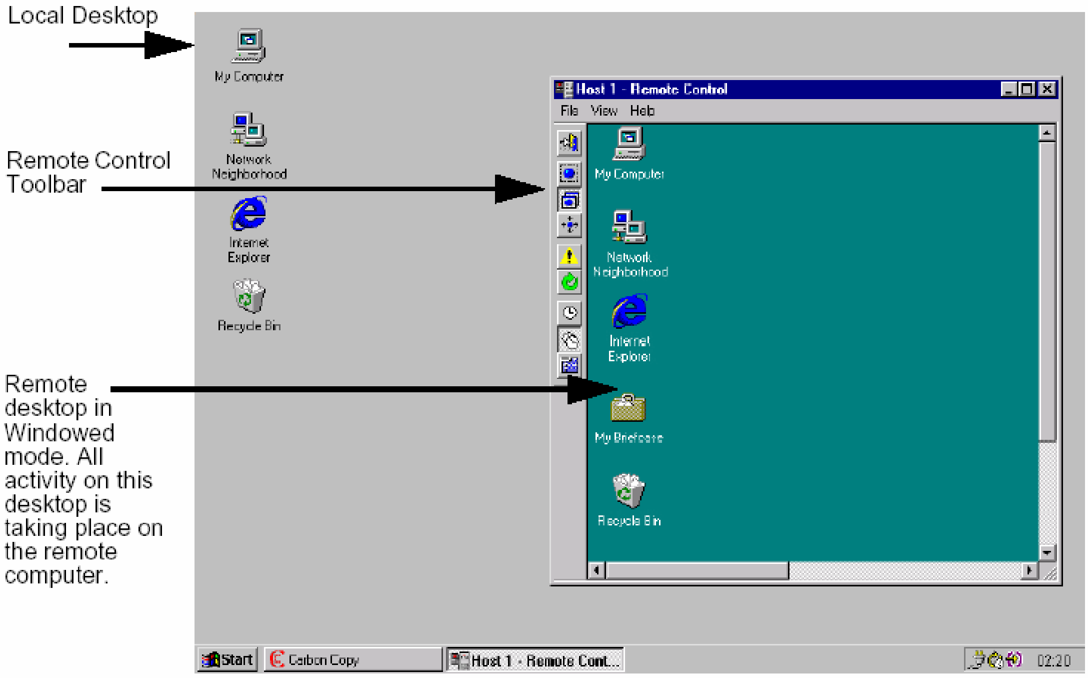


Figure 4.14 Carbon Copy application

The Computer initiating the Connection is called the Guest (also called the console), which is the viewer in VNC terms, and the computer that accepts the connection is called the Host (also called the client), which is the server in VNC terms.

Carbon Copy solution consists of three components: first, Carbon Copy solution Notification server components. Second, Carbon Copy solution Console: web-host console than can run up to 256 simultaneous connections, handheld console runs on Pocket PC 2000 or 2002, and window utility. Third, Carbon Copy solution client. To remotely control a client computer the first and the third component must be installed on that computer.

Carbon Copy has a feature to assign passwords to access selected areas of the client's user interface, it also login attempts and set the maximum number of attempts before connections is dropped. And as well as that Carbon Copy has a Locking feature where it locks all systems from connecting to the client after exceeding the allowed amount of failed login attempts or only lockout the offending systems. Another feature is operating in asynchronous mode where remote computer runs in real-time, while the local display scans the remote display and display changes, this is useful for graphic intensive or JAVA-based applications. Carbon Copy includes a set of security features: Data encryption, Connection Security, Directory restrictions and global security. If the computer is configured to receive connections and the login security is disabled any Carbon Copy user can connect the pc without providing any password. The user can either create user profiles with Carbon Copy or use Windows 2000/NT/XP user account.

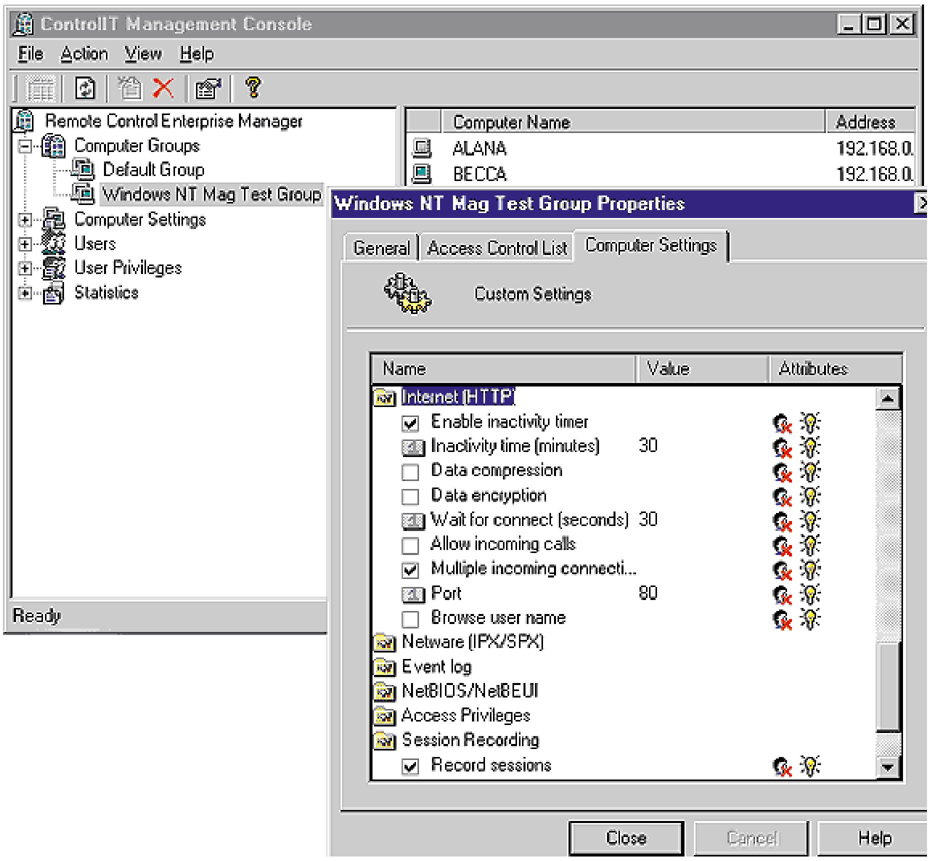
Carbon Copy comes in the following languages: English, French, German, Spanish, Italian, and Brazilian Portuguese.

Users of Carbon Copy are AOL, AT&T, BAE Systems, DELL, ERICSSON, FedEx Express, IBM, California state University and many more.

4.3.1.3 Unicenter Remote Control [132]

Unicenter Remote Control from Computer Associates includes technologies formerly available as Control*IT*, Control*IT* Enterprise Edition, Control*IT* Workgroup/Advanced Edition, Unicenter TNG Remote Control Option.

Unicenter calls the server "Host" and the client "viewer". The Unicenter remote control is shown in Figure 4.15



Computer Name	Address
ALANA	192.168.0.
BECCA	192.168.0.

Name	Value	Attributes
Internet (HTTP)		
☑ Enable inactivity timer		
Inactivity time (minutes)	30	
□ Data compression		
□ Data encryption		
Wait for connect (seconds)	30	
□ Allow incoming calls		
☑ Multiple incoming connecti...		
Port	80	
□ Browse user name		
Netware (IPX/SPX)		
Event log		
NetBIOS/NetBEUI		
Access Privileges		
Session Recording		
☑ Record sessions		

Figure 4.15 Unicenter Remote Control

This solution works on Windows XP, NT 4.0 and 2000. It supports Internet Explorer and Netscape Navigator where it allows full remote control from within a web browser. Remote Control is build on CA's Unicenter, so the user can integrate other management applications like Enterprise Discovery™, Real World Interface, Central alert and Event monitoring features and virus detection software to enhance the manageability of remote systems.

One of the interesting features is the autoPanning where if the host screen size is different from the viewer screen size it automatically adjusts the screen without the need to change the configuration settings.

Many features in this package allow it to be manageable such as centralized user management where administrators are able to change and manage access rights and remote control capabilities for a group of users and Session management where all the information about the Remote Control activities is stored in central logging and auditing system. Dynamic Enterprise Discovery: the system can find out systems on the network that is using CA remote Control solutions software and obtains different kind of information about activities on the remote host. Connect to multiple hosts where you can control several hosts simultaneously. This is beneficial in the case of a Help Desk where many inquiries can be answered. And passive monitoring where administrators/users can view the host desktop and environment without intruding on the work being commenced on it. Connect to multiple viewers: this is useful in a classroom scenario where the tutor can share his screen with his tutees to show them how to perform a certain task. Session record and session replay is supplied and there is a chat facility and can copy from and to the host machine.

In addition to that it offers a bidirectional file transfer facility (drag and drop operations) and a chat interface. Also it provides compression and encryption using 3DES. Unicenter remote control Provides strong management functions; integrates into Computer Associates' Unicenter TNG management software. However, it is complicated and difficult to manage and configure.

Some of the users of Unicenter remote control are Sichuan Mobile Telecommunications Company (Sichuan Mobile) and Radiodiffusion et de Télévision Française pour l'Outre-mer (RFO) – France's overseas public radio and TV network.

4.3.1.4 PcAnywhere [133]

Symantec pcAnywhere is a very popular and secure choice in remote control.

To use pcAnywhere two computers must run pcAnywhere, one of them has to be configured as a "host" which acts as a server and the other as "Remote" that acts as a client, the host configuration controls who can connect to the computer and what level of access the remote user should have.

The same computer can work as a host and client but this can be set at installation where host only or remote only can be chosen. There is a functionality called Packager which allows administrators to create, modify and build customized installation sets that contains only the features and settings to fit the needs of the user and suit your corporate environment. Although pcAnywhere has an ActiveX control version, the version resides in pcAnywhere's unsupported directory, and the product doesn't install this version by default or mention it in the documentation. The pcAnywhere manager is shown in Figure 4.16

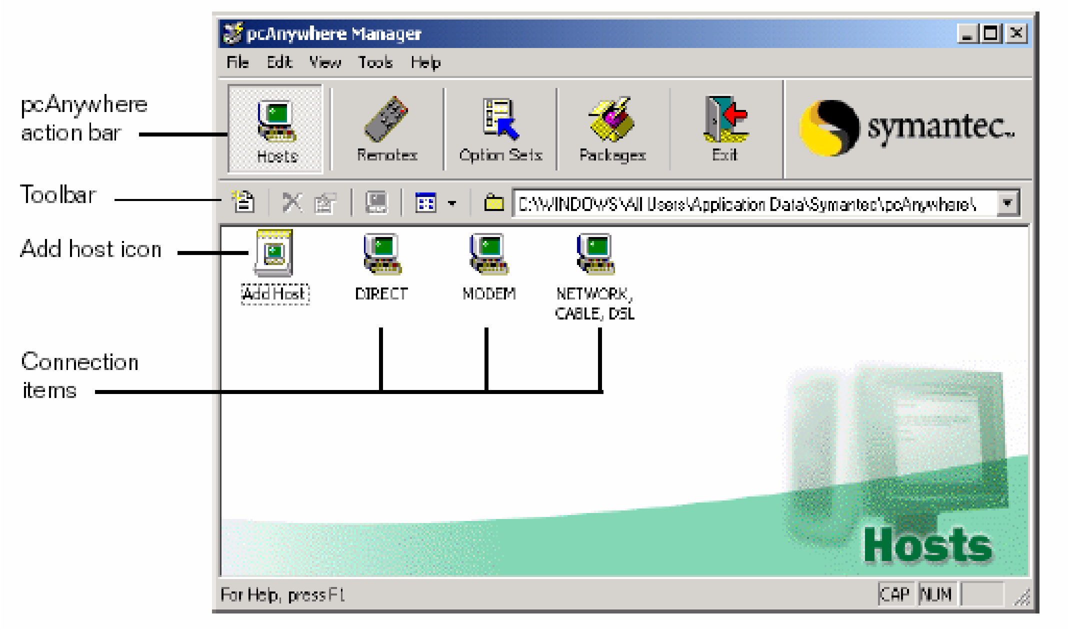


Figure 4.16 PcAnywhere Manager

Some of the Security features of pcAnywhere are Remote Access Primeter Scanner (RAPS) where this facility scans the network for unsecured pcAnywhere hosts and it can scan for other Remote Access products such as Carbon copy, NetMeeting, VNC, NetBus, Citrix server, X server and Terminal Server and then alert and give details about the vulnerable computer. It also provides three levels of encryption. The time limit can be set for a remote session and that will protect the host from malicious users' intents and you can restrict the remote user from performing certain tasks. One of the interesting features are blanking the PC screen after connection so any one at the host site will not be able to see what is going on the host monitor. Another feature is in the case of conferencing multiple users can connect to a single host where the first users controls the host and the other users can only view the activities on the host ; a user can connect to multiple hosts. And there is a logging utility to capture all failed attempts to connect to pcAnywhere due to incorrect serial number or unauthorised TCP/IP address including the TCP/IP address of the person trying to connect and the time of the attempt. pcAnywhere also has a status icon on the system tray that shows the status on who is connected to a computer, the computer name and the length of the connection. It also logs activities to monitor session activities and track performance issues. In addition to recording session it is possible to playback the visual recording of all activities performed on the local computer. Moreover, pcAnywhere has a File transfer utility.

4.3.2 Comparative Analysis

In general, two main factors that affect the performance of remote control software are: speed of the connection and display issues. When configuring a connection, a balance must be sought between security and high performance, because data encryption slows the performance, stronger encryption requires more resources to process and transfer the data.

Most of what you see on Windows is graphical user interface which means that a lot of graphics should be sent over the connection from the host computers to the remote computer and this slows performance, therefore, to minimise overhead the user can reduce the desktop resolution and number of colours on the display and disable wallpaper, backgrounds, and screen savers on the host. The user can also avoid using animation, use the Page Up and Page Down keys to scroll through documents, Disable the scroll wheel on the mouse and avoid performing remote control operations when transferring files to improve performance.

One of the desirable options is a remote control product that is operating over the Internet, including using existing dial-up or broadband Internet access providers. The product can either connect to an IP address or a computer name, which is not always the case with public accounts, which does not have a static IP address. [134]

VNC shares many powerful features with the other software, where a desktop can be shared by multiple users in addition to the ability to connect to multiple servers from the viewer.

Studying different remote control solutions lead us to find out many features that are not included in the original VNC and to look for either modified versions of VNC or try to integrate different tools. Netsupport, ControlIT, pcAnywhere and Carbon Copy let you transfer files between host and remote computers and record and play back sessions.

In addition to the case of having a corporation firewall which blocks access to any of its computers behind the firewall, to achieve the ability of using remote control software the following can be done: open up ports, encrypt traffic, implement dial-up connections into your private LAN, or use a directory server. Opening one or more ports on the firewall to allow the users to access specific hosts could be risky and messy for administrators. The use of web browsers using HTTP or IP protocol is very useful for remote control purposes to get around firewall issues where most networks don't block access on this port.

VNC makes use of a secured shell (SSH) program to provide an encrypted path between the server and the viewer (client) and to bypass firewalls which will be explained in chapter five.

PcAnywhere and VNC could restrict access by IP address or subnet. Carbon copy can limit access to particular directories on the host for all remote users.

After looking at the features represented in the four remote control solutions several weaknesses and missing features are deduced such as:

1. File transfer function: Ultra@vnc [135] implements a file transfer facility for windows platforms. Also, it is included in the commercial 4.x personal and enterprise edition of VNC.
2. Print redirection: When the user who is using the viewer wants a file existing on the remote server to be printed on his local machine. This is not a VNC issue however the user can overcome this problem by printing the document to a file and then transfer it to his machine and then print it or on the server you can redirect the printing to a networked shared printer.
3. Record and Replay session: have been developed but not integrated into VNC in an executable package. Although there is add on for UNIX in [58].
4. Blank server screen: this is not implemented in VNC but it is a desirable option which has been implemented in ultra@VNC [135].
5. Does not support multimedia: for example in some applications the user would like to listen on his viewer machine to the sound played on the server. VNCAudio[4] implements this capability for Linux.
6. Remote deployment of software: there is a script called Fastpush [136] that can remotely install WinVNC on to computers running NT, 2000, XP. In addition to VNC deployment tool developed by the original VNC team and is used to simplify the management of large corporate installations.
7. No text chat facilities: ultra@VNC [135] provides a chat facility
8. Encryption: Zvnc [137] is a secure VNC solution developed for Windows platforms and several different methods of encryption are discussed in chapter 5 and new VNC 4.x commercial personal and enterprise versions employ encryption.
9. Limited copy and paste support: VNC only supports copying and pasting of ASCII text in both directions, for X applications a program called xcutsel is used to copy the clipboard between different X methods. On Java applets cannot

[4] http://www.freshmeat.com

access the clipboard of the machine on which they are running, so the Java viewer has a clipboard button which shows a window displaying the contents of the remote clipboard which allow you to manipulate it locally.

10. Scalability: looking at the VNC source code in <vncserver.h> the maximum number of clients WinVNC can tolerate is 128. While at the Xvnc source code in Xvnc/programs/Xserver/include/misc, the maximum client number is 128 while it has a maximum number of 50 clients per session. This is not tested and still needs verification.

5 Chapter Five

Security of Thin Client and Advanced Settings

Security is the sum of all measures taken to prevent loss of any kind; a fundamentally secure system is one which no user has access to anything. However entirely secure systems are useless therefore the acceptance of a certain amount of security risk is required in order to provide usability. The goal of security management is to minimise the risk involved in providing the necessary level of system usability [138].

VNC covers many aspects of security with the flexibility to use third-party security in addition if required. Out of the box, VNC uses a random challenge-response system; this provides the basic authentication that allows a viewer to connect to a server, and the password is not sent over a network. However, once connected, all the traffic is unencrypted, so could be snooped with the right tools. VNC's documentation recommends that, if higher security is important, the VNC protocol is "tunnelled" via a more secure channel such a SSH (Secure SHell). SSH is easily and freely available in the UNIX community, and clients are available for Macs and Windows, however SSH servers for these platforms are not always freely available, and a cheaper alternative is to route the connection via a UNIX machine.

The classical trouble administrators may face with VNC is that it can posses a security risk in their network and this is either due to misconfiguration or a security weakness which if found by the attackers a possibility of gaining access to the network, hacking the network system and initiating attacks on the organisation and hence stealing important data which in many cases can lead to disastrous consequences.

VNC is unpopular among network administrators and security consultants because of the excessive control it offers that makes the network resources highly vulnerable.

This chapter covers different security aspects of authentication, access control, encryption, physical security and auditing. Also it will expose password related and network related risks of VNC. And I will propose security measures and also I have implemented encryption of the traffic using SSH and mention different methods of encryption that can be used with VNC too and finally will explain a practical tested way of bypassing firewalls by using Tunnelling or reverse connection.

5.1 Security Aspects of VNC

5.1.1 Authentication

Authentication is the process where a network user establishes a right to an identity or the right to use a name. There are large number of techniques that may be used to authenticate a user such as passwords, biometric techniques, smart cards and certificates [139]. It is also the technique by which a process verifies that its communication partner is who it is supposed to be and not a pretender therefore it confirms the identity of any user trying to logon to a domain or to access network resources. VNC uses a challenge-response mechanism for exchanging the password between the server and the viewer. The client authentication scheme is discussed in 5.3.2.2. Also in Ultra@vnc [135] and the new commercial VNC 4.x versions of Enterprise Edition, uses Windows authentication.

5.1.2 Access control

Once authentication is complete, authorisation is implemented through access control. When a user attempts to access any resources, a decision must be made whether the action should be permitted or denied. Authorisation is the process of determining whether an identity (plus a set of attributes associated with that identity) is permitted to perform some action, such as accessing a resource [139].

No access control mechanism is incorporated in VNC in the sense that the user has full control of the server once authorised. However there is a read-only mode where the viewer can only view the desktop without interfering or changing anything and on WinVNC also IP address authorisation can be implemented. Third party tools can be used to provide access control of accessing resources. On UNIX it depends on the permissions set to the user by the system administrator and TCP wrappers can be used to limit access to authorised users.

5.1.3 Encryption [140]

Encryption is defined as the process of converting a message into a cipher text by using a key so that the message appears to be nothing but gibberish. However, the intended recipient can apply the key to decrypt and read the message [140]. Cryptography for all communication channels is needed to ensure that an attacker cannot alter the usernames and passwords as well as securing sensitive files and applications over the networks.

There are two types of encryption:

1) Asymmetric encryption (Public key cryptography): is a method of encryption based on the fact of having public key and a private key.
2) Symmetric encryption: This algorithm uses the same key to encrypt and decrypt the message. This way is much faster and if it is used with keys of sufficient length can produce unbreakable cipher text.

No encryption mechanism is incorporated in VNC free edition. However VNC can be tunnelled via a secure channel such as SSH/VPN/SSL. Different encryption methods have been tested and will be discussed later in the chapter. However, 128-bit AES

Session Encryption & Tamper-Proofing is incorporated in the commercial VNC 4.x personal and enterprise edition.

5.1.4 Physical security

If an intruder can sit in front any of the PCs, Workstations or servers, he/she may be able to take control of the network. Moreover, the data on the machine will be insecure and exposed for damage or deletion. Hence, the insurance of securing the physical site of the machines is of major importance.

5.1.5 Auditing

Auditing is the process of tracking the activities of users by recording selected types of events in the security log of a server or workstation. In XVNC all the information about who connected to the machine is in ~/.vnc/xxx.log where xxx is the name of the host however nothing is mentioned about the actions or events that happened in his session. While in WinVnc it is in the WinVnc.log but it only records the last session before booting the machine and this information is replaced by the new VNC session and it only lists the clients connected to the server without listing any activities done on the server.

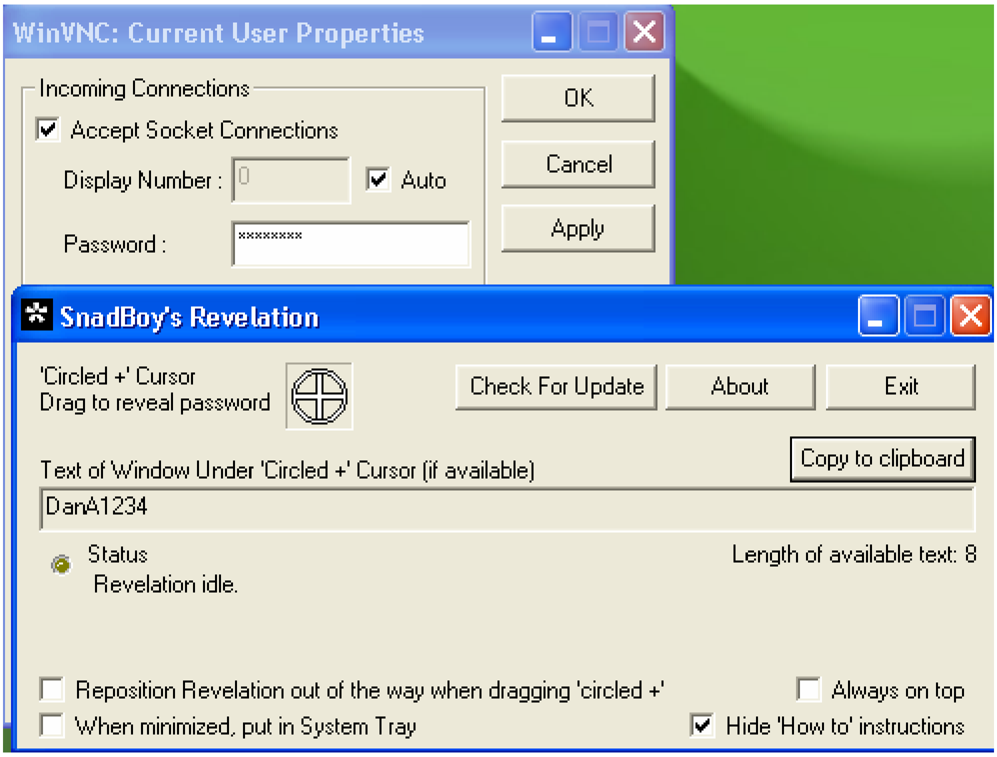
5.2 Security Risks

Many aspects of the security vulnerabilities were discussed in [141] such as: cleartext usernames and passwords, obfuscated password (using weak encryption algorithms like substituting), revealed passwords (pulled from the GUI either remotely or by copying the files locally), brute forcing VNC passwords and network eavesdropping. These aspects are discussed briefly here.

5.2.1 Password-related Risks

5.2.1.1 Revealed Passwords

Revelation from SnadBoy software [142] is 14K single executable reveals the passwords stored in the volatile memory space of many popular remote control software. Using Revelation, assuming you have the physical access to the winvnc's machine, you can "reveal" the password behind the asterisk simply by dragging the Revelation object over the password field. This security risk only applies on older versions of VNC up to 3.3.3rx and does not affect the newer versions. This can be shown in Figure 5.1 and Figure 5.2



WinVNC: Current User Properties

Incoming Connections

☑ Accept Socket Connections

Display Number : 0 ☑ Auto

Password : ********

SnadBoy's Revelation

'Circled +' Cursor
Drag to reveal password

Text of Window Under 'Circled +' Cursor (if available)

DanA1234

Status
Revelation idle.

Length of available text: 8

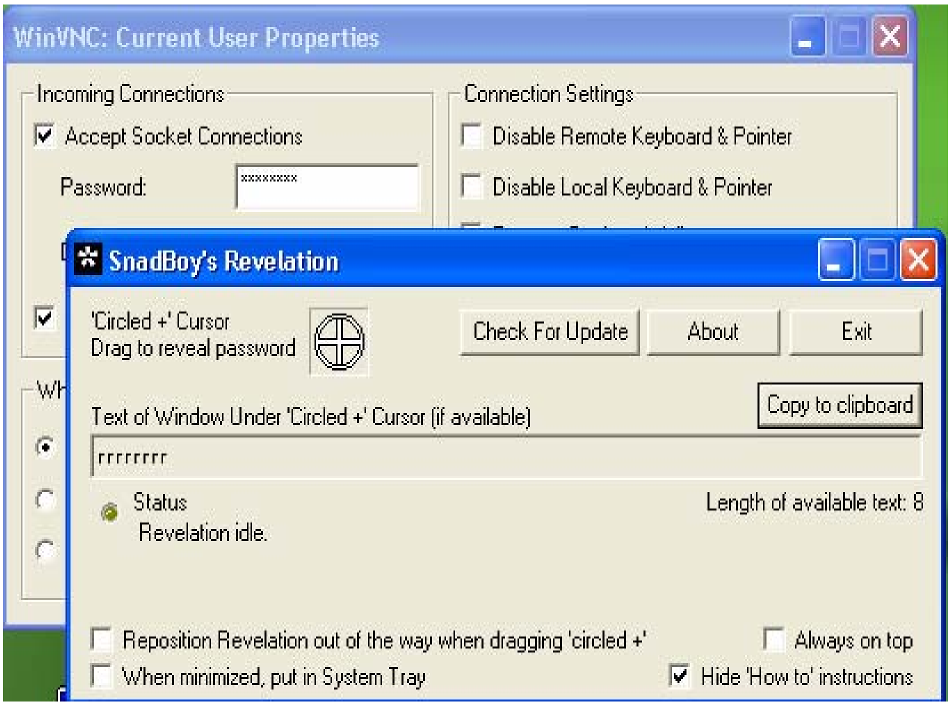
□ Reposition Revelation out of the way when dragging 'circled +'

□ Always on top

□ When minimized, put in System Tray

☑ Hide 'How to' instructions

Figure 5.1 Revealed VNC version 3.3.3rx password



WinVNC: Current User Properties

Incoming Connections

☑ Accept Socket Connections

Password: ********

Connection Settings

□ Disable Remote Keyboard & Pointer

□ Disable Local Keyboard & Pointer

SnadBoy's Revelation

'Circled +' Cursor
Drag to reveal password

Text of Window Under 'Circled +' Cursor (if available)

rrrrrrrr

Status
Revelation idle.

Length of available text: 8

□ Reposition Revelation out of the way when dragging 'circled +'

□ Always on top

□ When minimized, put in System Tray

☑ Hide 'How to' instructions

Figure 5.2 Unsuccessful revealed password in VNC version 3.3.4

5.2.1.2 Unsafe password field

A feature in the free edition of WinVNC is risky. This feature is that the password field can be changed without the need of supplying the old password so anyone who has the physical access to your PC can change your password so he/she could connect to it. This was overcome in the release of the improved VNC 4.x.

5.2.1.3 Weak WinVNC Password obfuscation

There are two main problems discussed in [143], the first problem discussed was fixed password length: where the VNC server will encrypt the password, but it will drop (nullify) all bytes after the 8th and the VNC password is stored in the registry this can be found in the source code of vncProperties.cpp:

Const char WINVNC_REGISTRY_KEY [] = "Software\\ORL\\WinVNC3";

Therefore the password is found at:

\HKEY_CURRENT_USER\Software\ORL\WinVNC3

The value of the password will be found as strange characters that are the encrypted version of the password that has been encrypted using Triple DES (Data Encryption Standard) and stored in the registry as hexadecimal.

The second problem was fixed encryption key:

VNC uses 3DES to encrypt the VNC server password. However, it uses a fixed key (23 82 107 6 35 78 88 7) every time a password is saved. This fixed key is available in the source code of vncauth.c as:

Unsigned char fixedkey[8]={23,82,107,6,35,78,88,7};

WinVNC assumes that the local file system is secure and this is the reason a fixed key was used, this is mentioned in vncauth.c .So it is simple to decrypt the password as far as we know the encryption key.

To decrypt the password a program called vncdec [144] can be used to recover the VNC password. I tested this patch on WindowsXP, Home Edition and compiled it using Div-C++ [145], which is a full-featured Integrated Development Environment (IDE) for the C/C++ programming language and its open source software. The password was taken as advised in the securiteam report from the registry and then placed in the vncdecrypt file and been compiled and checked and it really gives the decryption of the vncpassword.

It is important to note that since the attacker has the physical access to the server application he can change the password. However that would alert anyone who connects to the server and would trigger the increase of local security.

The solution is to secure your registry. This is accomplished by using one of the registry access control software. There are different utilities to protect registry entries by hiding them, or just restricting access to registry and some of them can set registry keys permissions and ownership.

Moreover, all versions of VNC since VNC 4.0 store sensitive information such as passwords with appropriate security permissions to avoid them being accessible to unauthorised users. As well as, using 2048-bit RSA for password encryption in the commercial versions of VNC.

5.2.2 Network-related Risks

5.2.2.1 Network eavesdropping on VNC

VNC listens for connections by opening specific ports on the host (server machine), by default its 5800…5899 and 5900…5999; however alternative ports can be chosen. Therefore, the attacker can start by using a port scanner to search for all computers on the network running VNC where port scanning [141] "is the process of connecting to TCP and UDP ports on the target system to determine what services are in a LISTENING state".

After discovering the machines or servers that run VNC, the attacker will start his second step of trying to gain access to these computers. So a main security hole is finding machines that run VNC server without any authentication requirement such as a null password field.

One of the good features of WinVNC is not allowing the server to run or to receive connections if the password field is empty. This is done by default and to change this feature the user needs to edit the registry value of **AuthRequired** to zero. Nevertheless this requires an experienced user and of course will have a reason to disable the password therefore this is not an issue in WinVNC. But on Xvnc if the user did not use the **vncserver** perl script, which automatically invokes the password, the server can run without a password and this can posses a security risk and can be started by a novice user. So administrators must always port scan their network and then try to connect to these machines to ensure nothing dangerous like that will take place.

Because VNC does not employ encryption between the server and the viewer after authentication it is possible to sniff the traffic between the two parties. Because of the fact of VNC being open source software, it is argued that it is easy to build a dedicated VNC sniffer. It is also argued that it is that it is harder to sniff VNC sessions than a telnet session because the traffic is compressed but it is possible. The password of VNC is exchanged using a challenge-response mechanism so it is not possible to sniff it; otherwise all other traffic is unencrypted so attackers can steal passwords concerning other systems that users log on [141].

The solution is Encrypting the traffic between the server and viewer using SSH to tunnel encrypted VNC sessions from the client to the server. And use TCP Wrapper for Xvnc and IP authorisation for WinVNC that provides access control on IP address basis.

5.2.2.2 Man in the middle attack

Another weakness found in [146] is the man in the middle attack and the vulnerable packages are VNC up to version 3.3.3 on all supported platforms. A design flaw in the client authentication mechanism allows an attacker to obtain legit credentials from a valid client in order to gain unauthorized access to the server.

Firstly the client authentication will be explained and then a Man in the middle attack implementation will be described.

The client authentication mechanism is shown in Figure 5.3 and explained here: when the VNC client tries to connect to the VNC server and the following protocol is used [146]:

1- A DES key (k) (password) is shared by both ends and used for the challenge response.

2- The client connects to the server and they exchange software/protocol version information.

3- The server generates a 16-byte challenge and sends it to the client.

4- The client encrypts the received challenge with (k) and sends the result (rc) to the server.

5- The server encrypts the challenge with (k) and compares the result (rs) with (rc) received from the client.

6- If rc==rs the client is authenticated and access to the server is granted

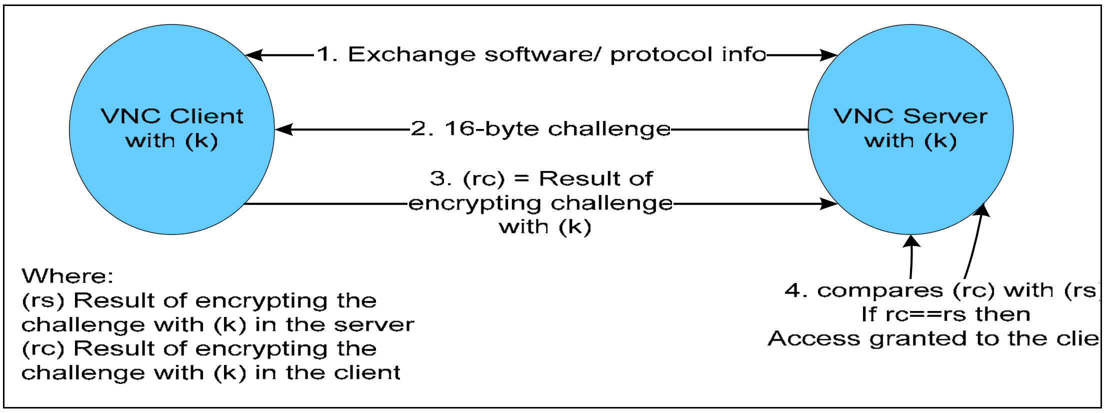

VNC Client
with (k)
1. Exchange software/ protocol info
2. 16-byte challenge
3. (rc) = Result of
encrypting challenge
with (k)
VNC Server
with (k)
Where:
(rs) Result of encrypting the
challenge with (k) in the server
(rc) Result of encrypting the
challenge with (k) in the client
4. compares (rc) with (rs
If rc==rs then
Access granted to the clie

Figure 5.3 Client/server authentication mechanism

Now if an attacker has access to the data flowing between the VNC client and the VNC server and is able to modify the data, the following man in the middle attack, which is shown in Figure 5.4, will be performed:

1- The attacker connects the vnc server and both parties exchange software/protocol information.

2- The server generated a 16-byte challenge (r1) and sends it to the attacker. Now there is a connection between the attacker and the vnc server with the authentication pending a response to the server.

3- The attacker waits for a connection from a legitimate vnc client connecting to the vnc server.

4- Now the server generates a 16-byte challenge (r2) and sends it to the vnc client.

5- Now the attacker manipulates the data travelling from the vnc client to the vnc server and replaces (r2) with (r1).

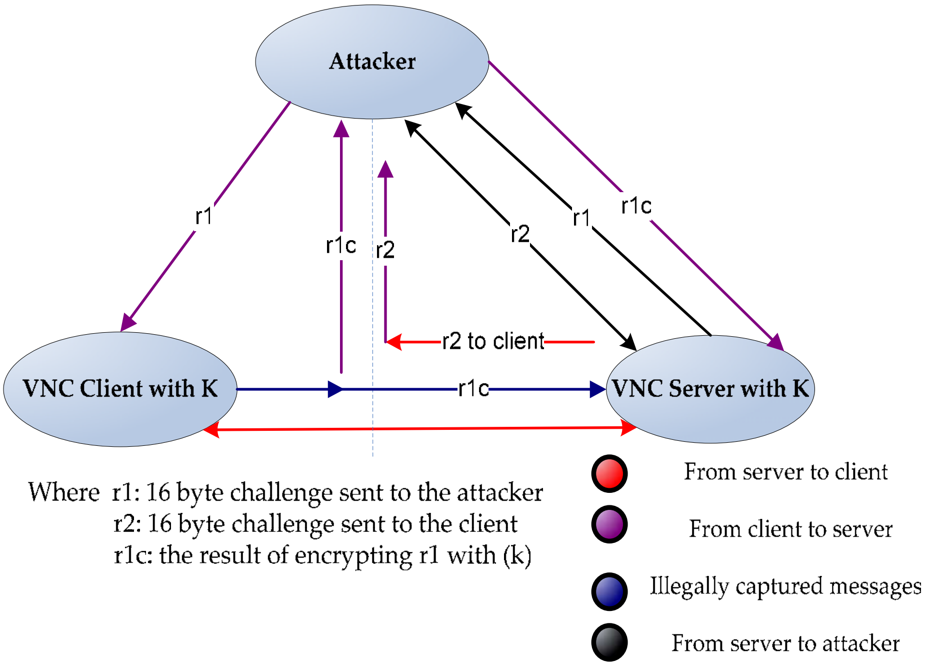
6- vnc client receives (r1) and encrypts it with (k) and sends the result (r1c) to the vnc server.

7- The attacker captures (r1c) sent to the server and it uses it as a response to its own pending connection.

8- Now the vnc server receives two (r1c) responses from the attacker and from the vnc client. It encrypts the challenges (r1 and r2) with (k) and compares the results (r1s and r2s) against the received responses.

9- For the legit vnc client connection (r2s!= r1c) and access is denied.

10- For the attacker connection (r1s==r1c) and access is granted.



Where r1: 16 byte challenge sent to the attacker
r2: 16 byte challenge sent to the client
r1c: the result of encrypting r1 with (k)

Figure 5.4 Man in the middle attack

5.2.2.3 Remote deployment of WinVnc

It is easy to install VNC on Windows NT over a remote network connection all you need to do is to edit a single line in the Registry and install the VNC service from the command prompt (the attackers can use older versions than 3.3.2 to ensure the stealthy mode whereas in newer versions a VNC icon will show in the system tray which denies the stealthy nature attackers wants to achieve).

An advantage is whether VNC is of an old or new version it is always presented in the Process List (found by using Ctrl +Alt + Delete simultaneously) and is always called WinVNC.exe.

5.3 Security Solutions

5.3.1 WinVNC Security Precautions

A secure solution is suggested in order to obtain a secure VNC session by manipulating the advanced settings of VNC.

The subsequent options are invoked by editing the Registry. To open the Registry: Start-> Run-> regedit

1. Users should not be authorized to view the properties dialog shown in Figure 4.1 and hence would not be able to change any settings including the password. This can be achieved by changing a property in the registry: Local per-user setting:

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\<*username*>

The DWORD **AllowProperties** should be set to zero, where the user is not allowed to view the properties dialog and hence cannot change any settings.

2. Users should not be allowed to close down the WinVNC server. This is done by setting

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\<*username*>

The DWORD **AllowShutdown** should be set to zero, where the user is not allowed to close down WinVNC.

3. One of the strong points of WinVNC is that by default, it will not accept incoming connections unless the server has the password field set to a non-null value.

4. One of the interesting features in WinVNC is it gives the ability to restrict access to your PC based on IP address basis and the user can either:

1) Allow the authorised IP address to connect to your machine quietly.
2) Whenever anyone is trying to connect to machine a dialog box will appear to inform the machine user of the attempt and gives you the choice to accept or reject the session.

Editing and creating few registry keys can achieve those security requirements. That can be accomplished by doing the following steps:

A new registry key must be created as a string value and named **AuthHosts** is a Local machine specific setting in

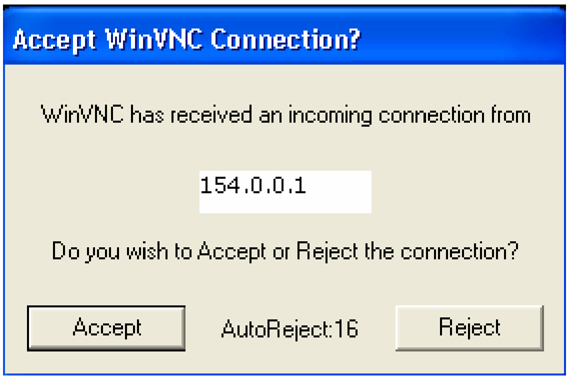
HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\

This key is used to specify a set of IP address templates which incoming connections must match in order to be accepted.

The following template is used to allow only a set of IP addresses that start with 168.40.69 to connect to your machine:

-:168.40.69

Terms beginning with the "?" character is treated by default as indicating hosts from whom connections must be accepted at the server side via a dialog box like Figure 5.5:



Accept WinVNC Connection?

WinVNC has received an incoming connection from

154.0.0.1

Do you wish to Accept or Reject the connection?

Figure 5.5 Accept Dialog

This can be done by using the following string in the AuthHosts:

-:?154.0.0

Other settings are associated with **AuthHosts** such as **QuerySetting** and **QueryTimout**. Unlike AuthHosts, QuerySetting and QueryTimeout are DWORD registry values.

The user should edit the **Local & global per-user setting** in order for the changes to take affect**:**

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\<*username*>

HKEY_CURRENT_USER\Software\ORL\WinVNC3

QuerySetting expresses the level of paranoia the user chooses which complies with AuthHosts ranging from 0 to 4, 0 being the most trusting and 4 the most secure.

QueryTimeout indicated the number of seconds the accept dialog will show before rejecting the connecting.

It is recommended to use the accept dialog boxes in a call centre scenario where the user must be informed before the administrator connects to his/her machine. So the call centre staff can implement the AuthHosts by setting it to only accept their machines IP address and to prompt for the authorization of the user which solved any privacy complaints the users can claim to have.

One of the weaknesses of AuthHosts is it does not recognise the Java viewer even if it is used from the authorised IP address range so for security it is recommended to use the standard VNC viewer.

5. To allow many connections to the same desktop to allow sharing, using WinVNC in a CSCW style, a property **ConnectPriority** which is a Local machine specific setting

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\

ConnectPriority is a DWORD:

0= Disconnecting all current existing connected clients

1= Do not disconnect any existing connections (share mode)

2=Refuse any new connections (secure to avoid anyone interrupting your session)

6- In order to know who is connecting to your desktop, a screen can be displayed which shows the IP address of the machine that connects to your desktop.

This can be achieved by setting the following:

Local machine-specific setting:

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3

You should create two DWORD registry keys: **DebugLevel** , **DebugMode**

DebugMode indicates which logging methods to use,

[1 = MSVC debugger]

2 = Output to log file Winvnc.log in the WinVNC directory

4 = Output to a console window, displayed on-screen

Any combination of the above values may be used. Example, DebugMode=6 will cause output to be sent both to the WinVNC.log file and to the console window on the desktop.

DebugLevel indicates how much debug information to present. Any positive integer is valid. Zero indicates that no debugging information should be produced and is the default. A value of around 10-12 will cause full debugging output to be produced.

7. Another way to know who is connected to the VNC server a command line utility called VNCWHO is used [111]. This program may be run on any Microsoft Windows operating systems (Windows XP/2000/NT and 9x/Me). VNCWHO lists the IP addresses of any PCs or Workstations connected to the WinVNC.

Or you can launch the command prompt and type the following command:

```
C:\>netstat –a
```

Which lists all the TCP listening sockets and on TCP sockets 58xx and 59xx the IP address of any connected machines is listed.

8. VNC allows the user to have control of the whole desktop so a mean of access control of the folders must be implemented according to the policy the owner of the desktop prefers.

Many utilities are developed to hide any folder completely and securely, the user just needs to specify the folders and activate the utility and they will be hidden from unauthorised access. Files held in the concealed folders cannot be manipulated or modified, nor can they be deleted or even opened.

9. If the owner of a Window machine found a VNC service on his machine without his authorisation he should kill the service and run the Java program (1) in **Appendix B** on his machine to log any attempt to connect to port 5900 with the IP address of the attacker and the date and time.

Note: It is worth mentioning that in the new VNC versions of 4.x, contain most of the registry keys needed to be changed here as a part of the Winvnc properties window.

5.3.2 Encrypting traffic

Several encryption methods are explained

5.3.2.1 SSH

SSH is a powerful method of performing client authentication and safeguarding multiple service sessions between two systems. Systems running SSH listen on port 22 for incoming connection requests. When two systems running SSH establish a connection, they validate each other's credentials by performing a digital certificate exchange using RSA. Once the credentials for each system have been validated, 3DES is used to encrypt all information that is exchanged between the two systems. The two hosts authenticate each other in the course of the communication session and periodically change encryption keys. This process helps to ensure that brute force or playback attacks are not effective. SSH encapsulates insecure sessions to ensure that no cleartext information is visible [138].

In addition to the direct access provided by SSH, it also provides a technique called "port forwarding" or "Tunnelling". This can provide indirect access to other services that would not normally encrypt data during transmission, in our case tunnelling VNC's traffic to encrypt its sessions. SSH implements compression for the traffic it encrypts too.

A. SSH Implementation

I have implemented an approach of encryption for VNC's traffic using SSH and this will be discussed in the following section:

1. SSH Server

Firstly, SSH2 package is downloaded from ftp://ftp.ssh.com on Solaris SPARC 8:

1- uncompress the package , in our case its:

gzip –dc ssh-3.2.3.tar.gz | **tar xvpf –**

2- ./configure

3- make

4- su as a root and enter password and then type csh

5- make install

6- to enable ssh1 compatibility add two line in **sshd2_config** placed at "**/etc/ssh2**":

```
Ssh1Compatibility              yes
Sshd1Path                      /usr/local/sbin/sshd1
```

Then add 2 lines in **ssh2_config** placed at "**/etc/ssh2**":

```
Ssh1Compatibility              yes
Ssh1Path                 /usr/local/sbin/ssh1
```

7- to use ssh as a service the following files must be edited, **/etc/inetd.conf** and **/etc/services** and then send a hangup to inetd:

```
ps –ef | grep inetd (to find inetd's PID)
Kill –HUP PID
```

And then start inetd and ssh will be working as service

Notes:

1- to enforce a different compiler such as gcc type the following:

env Cc=/usr/local/bin/gcc ./configure and/or make

2- to save yourself typing the paths to the commands, they have to be set in your path:

set path = (/usr/local/bin /usr/ccs/bin $path) where /usr/local/bin is the directory where the gcc and /usr/ccs/bin is where **make** command.

2. SSH Clients

Second, now the ssh server is up and running and any ssh client can connect to it. The following implementations were tested using Solaris 2.6 for the UNIX machines and Windows XP for the Windows machines.

a) Windows SSH client

PuTTY [147] is a free SSH client implementation for Windows and very easy to use. In order to connect to the SSH server on UNIX, you should run PuTTY and enter the machine name or IP address of the UNIX workstation in the Host name field in Figure 5.7 and choose the SSH option beneath it. Figure 5.6 explains an arrangement where a Window client wants to view either a UNIX server or a Windows server.

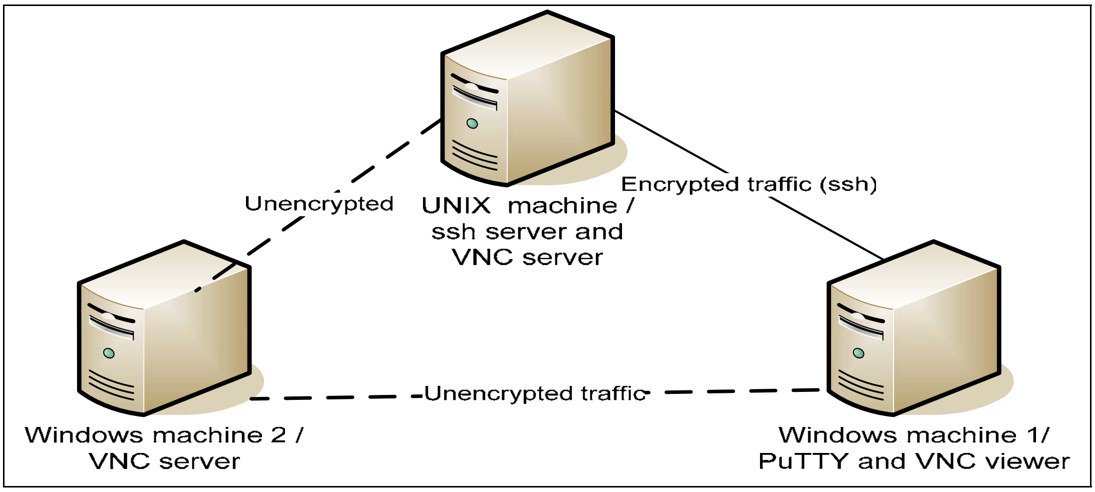
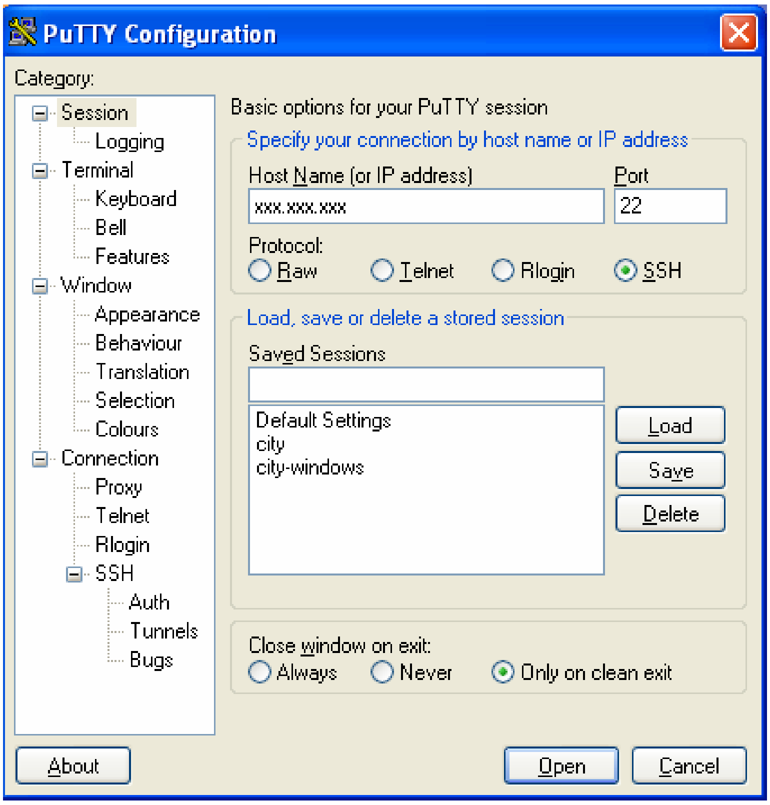


Figure 5.6 SSH implementation-Windows client



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): xxx.xxx.xxx

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

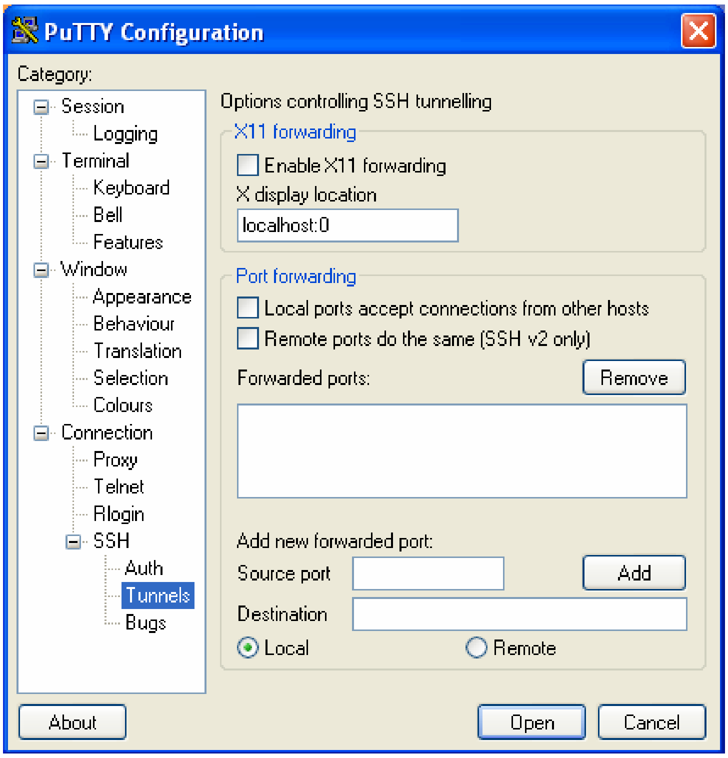
Load, save or delete a stored session

Saved Sessions:

Default Settings
city
city-windows

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 5.7 PuTTY Configuration-session details



Options controlling SSH tunnelling

X11 forwarding

☐ Enable X11 forwarding

X display location: localhost:0

Port forwarding

☐ Local ports accept connections from other hosts

☐ Remote ports do the same (SSH v2 only)

Forwarded ports:

Add new forwarded port:

Source port

Destination

● Local ○ Remote

Figure 5.8 PuTTY Configuration-Tunnels

In Figure 5.8 in:

Source port: enter the local port number you want the ssh session to forward to the target machine such as 59xx where xx is the display number.

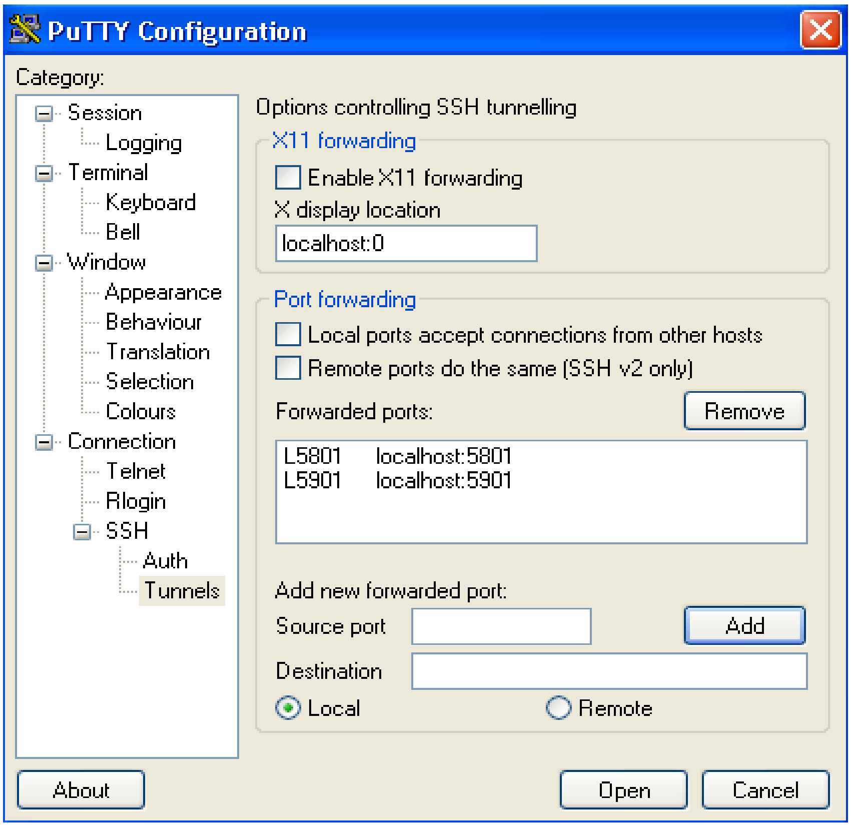
In **Destination**: the IP address of the machine the VNC session is running on: port number in the following form: xxx.xxx.xxx: 59yy where yy is the display number of the vnc server. This is **Local** port forwarding.

Looking at Figure 5.6, there are two scenarios:

1) Connection from Windows client to a UNIX VNC server

Windows machine 1 wants to secure its traffic to the VNC server running on the UNIX machine that runs the SSH server. The user on Windows 1 runs putty and enters the IP address of the UNIX machine in the **host name** and in the port forwarding section he specifies in the **Source port** the local port number (on Windows 1) he wants to forward to the target machine such as **59xx** and in the **Destination** field he enters: **localhost:59yy** where 59yy is the port number of the vnc server running on the UNIX machine and then press Add.

To use the Java viewer you should add another option in addition to the last one the **Source port** should be 58xx and the **Destination** is localhost:58xx and then press add. In this case both **Source port** 59xx and 58xx should be equal to **Destination ports** like Figure 5.9



Options controlling SSH tunnelling

X11 forwarding

□ Enable X11 forwarding

X display location: localhost:0

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH v2 only)

Forwarded ports:

```
L5801     localhost:5801
L5901     localhost:5901
```

Add new forwarded port:

Source port

Destination

● Local ○ Remote

Figure 5.9 Putty configurations for standard and Java viewer

It only works if you are using the same ports on the **Source** and **Destination**. These results have been obtained after a series of tests using different ports and by looking at the original index.vnc (it starts the applet):

```
<!-- index.vnc - -->
<HTML>
<TITLE>
$USER's $DESKTOP desktop ($DISPLAY)
</TITLE>
<APPLET CODE=vncviewer.class ARCHIVE=vncviewer.jar
WIDTH=$APPLETWIDTH HEIGHT=$APPLETHEIGHT>
<param name=PORT value=$PORT>
</APPLET>
</HTML>
```

The variable $PORT is replaced through the port by Xvnc and when you are tunnelling through SSH the port for the applet will be the Source port because the applet must connect to your local port on the client's machine, while XVNC replaces the port with the destination port that's why both ports must be the same.

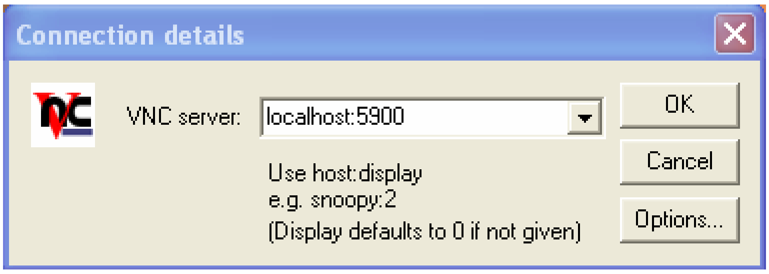
2) Connection from Windows client to Windows VNC server

Windows machine 1 wants to connect securely to Windows machine 2 running a VNC server. This can be achieved through the UNIX machine running the SSH server. Now the user on Windows 1 runs PuTTY and enters the IP address of the UNIX machine that runs the SSH server in the **host name** and in the port forwarding section he specifies in the **Source port** the local port number he wants to forward to the vnc server running on Windows machine 2 **59xx** and in the **Destination** field he enters the IP address of the Windows machine 2 **Windows2 IP address: 59yy**

To use the Java viewer follow the same technique as in scenario 1.

- Using VNC viewer on the Windows client machine

In both scenarios after connecting to the SSH server the user on his local machine runs the VNC viewer and enters in the VNC server: localhost: 59xx where 59xx is the source port in the PuTTY configuration such as Figure 5.10.



Connection details

VNC server: localhost:5900

Use host:display
e.g. snoopy:2
(Display defaults to 0 if not given)

Figure 5.10 VNC viewer with ssh

And in the browser:

http://localhost:58xx

b) UNIX SSH clients

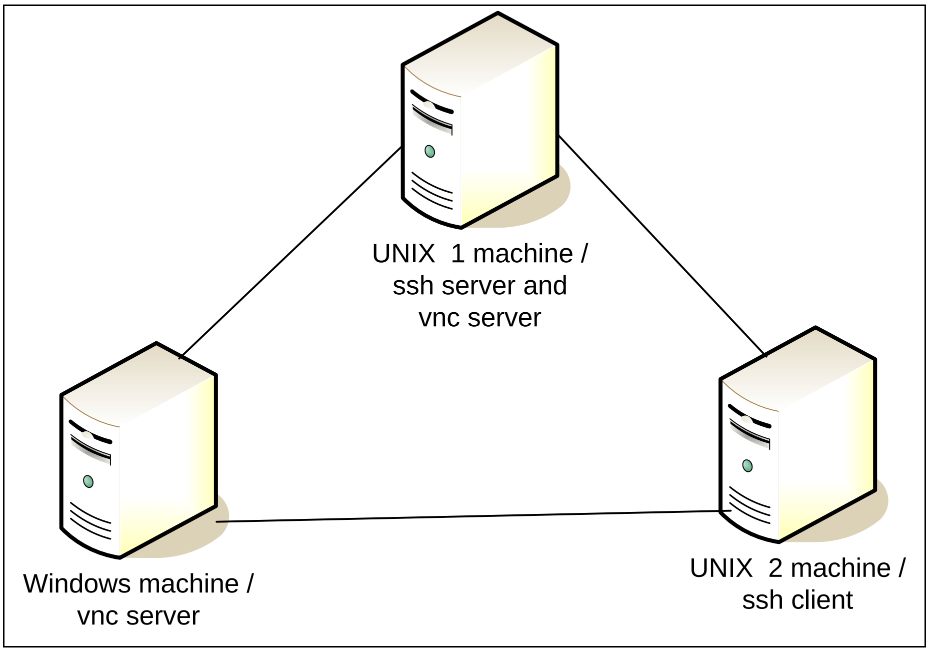


Figure 5.11 SSH implemetation-UNIX client

If the client was a UNIX machine like it is shown in Figure 5.11 that have a SSH client the following command is used to achieve the port forwarding. To create a TCP/IP port-forwarding channel, which listens for connections on the localhost, use the following command:

```
ssh -L local-port:remote-hostname:remote-port username@hostname
```

1) Connection from UNIX client to UNIX VNC server

UNIX 2 wants to have a secure vnc session running on UNIX 1:

```
ssh –L sourceport:localhost:vnc-port username@sshserver
```

where sourceport corresponds to the local port on UNIX 2 machine.

vnc-port : corresponds to the port of the vnc server running on UNIX 1.

username@sshserver: is your account on the UNIX 1 @ ip address of UNIX 1 machine.

2) Connection from UNIX client to Windows VNC server

UNIX 2 machine wants to connect securely to Windows machine running a VNC server. This can be achieved through the UNIX 1 machine running the SSH server:

```
ssh –L sourceport:vncservermachine:vnc-port username@sshserver
```

Where sourceport corresponds to the local port on UNIX 2 machine.

vnc-port corresponds to the port of the vnc server running on Windows machine.

username@sshserver: is your account on the UNIX 1 @ ip address of UNIX 1 machine.

And then on the local machine the following must be typed to run an encrypted VNC session:

```
vncviewer localhost:sourceport –encoding copyrect hextile
```

By default, when the viewer connects to a server on the local machine, it uses VNC's 'raw' pixel encoding because this generally gives better performance for local access. If this 'server' is actually an SSHD redirecting the data to another machine, you probably want to override this using the -hextile option to the viewer, or you will send a lot more data over the network than is necessary. (In the latest versions of the viewer, use -encodings "copyrect hextile").

Java VNC: In order to encrypt the Java equivalent of the standard VNC viewer the following command:

```
ssh –L 58xx,59xx:vncservermachine:58yy,59yy username@sshserver
```

Where x and y range from 0 to 99 and xx can be equal yy

And then http://localhost:58xx in the browser

B. Disabling non-encrypted connections

The following section is to explain how to achieve maximum security if the user would like only encrypted connections to connect to VNC and decides to drop other unencrypted connection.

1) To disable non-encrypted connections on Windows 2 machine

If you allow **AllowLoopback** Access and deny all hosts in **AuthHosts** except 127.0.0.1, you can limit network access only to those being forwarded by SSH. This will also allow unencrypted connections from the local host but that probably is not a problem. **AllowLoopback** is a Local machine-specific setting in

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\

And should be set to 1 to allow local loopback connections.

Authhosts is explained previously. The other method is set the registry entry of **LoopbackOnly** to 1 and this will cause WinVNC to only accept local connections - this overrides the AllowLoopback and AuthHosts settings. Setting this entry to zero causes WinVNC to accept connections on any adapter and is the default setting. Local machine-specific setting in HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\

2) To disable non-encrypted connections to UNIX machine

The user should configure the TCP wrappers appropriately, to lock down access to VNC except from localhost, in

/etc/host.deny:

```
ALL: ALL: /usr/bin/mailx –s "%s: connection attempt from %a \
yall@yourdomain.com
```

/etc/host.allow:

```
#Access for ssh currently unrestricted
sshd sshd2 : ALL
#Access for vnc restricted to localhost (remote via ssh tunnel)
/usr/local/bin/Xvnc –query localhost –once –localhost :localhost
```

The user can limit access to specific IP addresses using TCP wrappers too.

5.3.2.2 VPN

VPN, or Virtual Private Network, is a cryptosystem that allows you to secure your data as it travels over an insecure network such as the Internet. While this may sound similar to the SSH cryptosystem, VPNs have a different purpose. SSH was designed to allow a user to login securely to and remotely administer another computer. A VPN is designed to allow a user to access transparently the resources of a network. As far as the user is concerned, she will be able to do anything she normally would be able to do, even when she is away from the network. Because of this, VPNs are popular with telecommuters and with offices that need to share resources over physically separate locations.

There are many VPN purposes and three types are classified to address these purposes [148]:

1) Access VPN: a VPN used to connect to the network over a shared medium like the Internet. People dialling the modem on their PC and connecting to a modem at work are crossing the shared medium of the public telephone system.
2) Intranet VPN: a VPN used to connect two trusted locations to each other over a dedicated connection.
3) Extranet VPN: a VPN used to connect untrusted locations to each other over a dedicated connection.

The VPN is a private network that uses a public network (usually the Internet) to connect remote sites or users together .VNC can be tunnelled through a VPN to provide encryption and security for the traffic.

5.3.2.3 Stunnel

Stunnel [149] is a program that allows the encryption of arbitrary TCP connections inside SSL and available on both Windows and UNIX. The user needs to download stunnel and OpenSSL on both the VNC server and client computers. The implementation of Stunnel and VNC on Windows is fully explained in [150]. SSL is a protocol created by Netscape communications to provide RSA encryption at the session layer of the OSI model [148] it enables encrypted and authenticated communication across the Internet.

5.3.2.4 Zebedee

Zebedee [151] is a simple program to establish an encrypted, compressed "tunnel" for TCP/IP or UDP data transfer between two systems. This allows traffic such as telnet, ftp and X and VNC to be protected from snooping as well as potentially gaining performance over low-bandwidth networks from compression. Zebedee is named after its three main components: **Z**lib compression , **B**lowfish encryption and **D**iffie-Hellman key agreement. Zebedee supports the use of public and private keys to check the identity of clients before allowing utilising the zebedee secure tunnel and it restricts access to specific users to achieve strong authentication.

Using this feature with VNC provides layering your security where user's identity connecting to the Zebedee server is first checked based on the private and public keys system and then the VNC server will ask for the server's password.

zVnc [137] is a Windows only implementation based on VNC and Zebedee. zVnc is a development by Dave Dyer and it includes zebedee compression/encryption. The components of Zvnc are: VNC as a main component, ZeBeDee as a secondary component, Blowfish encryption extracted from OpenSSL, BZip compression library and Zlib compression. One of the bugs in Zvnc is it does not incorporate the vnc's -listen mode. It is possible to use ZeBeDee with VNC, but doing so requires additional configuration of both client and server, and inherently is less efficient and more prone to error than using VNC alone. Zvnc is only Windows specific.

zVnc integrates encryption and compression into VNC, while remaining compatible with both ordinary VNC and with configurations where VNC and ZeBeDee were used in combination. Perhaps surprisingly, the compression component of the combination is a significant improvement over VNC alone. zVnc is based on **VNC 3.3.3R9** and is compatible with it so instead of running WinVNC the user run ZWinVNC , ZWinVNC will accept connections on port 5900+n from any standard Vncviewer. It will also accept encrypted, compressed connections from ZVncviewer (or from ZeBeDee) on port 6000+n.

To connect to ZWinVNC from a normal Vncviewer tunnelling through ZeBeDee, use -T 6000 to force the local ZeBeDee to connect on port 6000 instead of 11965. In this case, the port redirection specified by ZeBeDee is ignored (and irrelevant).

Instead of using vncviewer you should Run ZVncviewer. On one hand to connect *insecurely*, use *hostname*, on the other hand to connect *securely* to a ZVNC server, use *hostname:6000* instead of *hostname.* Any port number 6000 or greater will attempt a secure connection.

To connect to a remote ZeBeDee tunnelling to a standard VNC server, use *hostname:11965* . 11965 is the socket ZeBeDee servers normally listen for connections. ZVncviewer will appear to be another copy of ZeBeDee requesting a tunnel to port 5900 which is the port WinVNC normally uses.

5.4 Overcoming Firewalls

Firewalls are used to create security checkpoints at the boundaries of private networks. Inspection of all packets passing between the private network and the Internet is made at these checkpoints and the determination whether to pass or drop the packets depends on how they match the policy rules programmed into the firewall. Firewalls sit on the borders of the private network. Firewalls have three methods for functioning:

1- Packet Filtering: declines TCP/IP packets from unauthorised hosts and any connection attempts to unauthorised services.

2- Network Address Translation (NAT): also known as *IP masquerading,* this type translates the IP addresses of internal hosts to hide them from outside monitoring.

3- Proxy services: makes high-level application connections on behalf of internal hosts in order to break the Network layer connection between the internal and external hosts.

The firewall performs two main security services:

a) Encrypted authentication: where the user identifies himself to the firewall in order to gain access to the private network from the external network

b) Virtual private networking: establishes a secure connection between two private networks over a public medium like the Internet [152].

5.4.1 By Tunnelling

I have preformed many attempts to try to connect to a VNC service running within the private network from outside the firewall such as it is demonstrated in Figure 5.12 using the IP address of the machines but all attempts were unsuccessful due to the firewall protecting it.

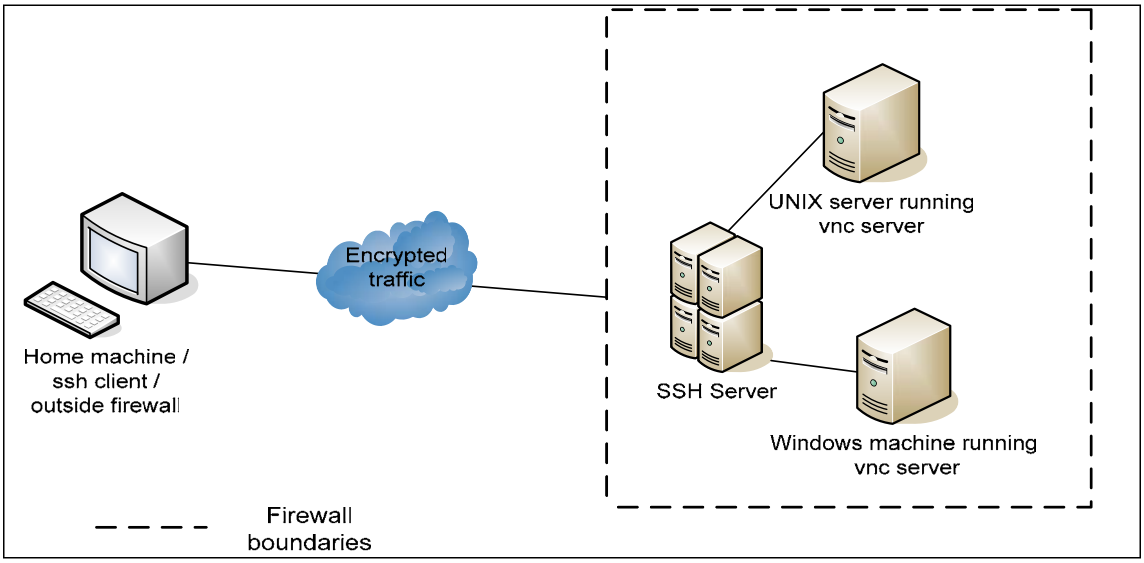


Figure 5.12 Bypassing firewalls

To bypass a firewall protecting a network using tunnelling, the firewall must have an open SSH port usually on port 22, where the student or the employee can connect from his/her machine to the SSH port using a SSH client and perform port forwarding to connect to a VNC server running in the internal network.

I have tested this arrangement and it fully worked and this can be accomplished by doing the following:

On my home machine which is a Windows machine, I used PuTTY to connect to the firewall protected server as explained in 5.4.2.1 where the **Host name** is the name of the firewall protected server, the **Source port** is the local port on home machine, the **Destination** is the IP address of the machine inside the private network that is running the VNC server. In Figure 5.12 if the home machine wants to connect to WinVNC running on Windows running internally, the **Destination** should be the IP address of the Windows PC and the port number is the port number on which VNC server is listening for connection on. And if the home user wanted to connect to the UNIX machine the only thing he/she should replace is the IP address and port number in **Destination.**

And then using the VNC viewer, type:

Localhost:59xx 59xx is the source port ,

or using the web browser type:

http://localhost:58xx 58xx is the source port

These methods only allow legitimate users to connect to the ssh server because it asks for an authorised account of the user to connect to it. In addition to that, the user must know the IP address of the PC or workstation he/she wants to connect to within the network in order to connect to it. This ensures that only authorised users can connect to the VNC servers where three levels of security is incorporated, first ssh server authentication, second, knowledge of machines' IP addresses and finally, VNC authentication.

If the Home machine was a UNIX machine the following command would achieve the same goal as the Windows machine in the last paragraph:

```
ssh –L sourceport:vncserver:vnc-port username@universitysshserver
```

and then

```
vncviewer :localhost
```

Many other aspects can be found in the following references: [153]-[154]

5.4.2 By Reverse Connection

To overcome a firewall a reverse connection can be initiated by the VNC server behind the firewall to the computer outside the firewall that runs the VNC viewer in a listening mode. This method is explained in detail in chapter 4.

6 Chapter Six

Applications and uses of VNC

This chapter outlines firstly the applications of VNC such as: Computer Supported Cooperative Work, computer support, computer training, virtual laboratories, distance learning, controlling home appliances, and using VNC with PDA and mobile phones and phones. Secondly, the chapter outlines several prototypes of a VNC based environment for collaboration and videoconferencing.

6.1 VNC Applications

6.1.1 Remote access, computer support and computer training

Many universities around the world are using VNC to facilitate remote access to their students and staff. There are different implementations of VNC in universities; one of them is to provide their users the access to all the graphical applications available under UNIX, for example the Linux operating system, without the need to install Linux to their PC. VNC is implemented for users, who have a computer accounts on the university's central UNIX server in order to run a *UNIX desktop environment* via a client on their desktop PC.

The mechanism of using VNC is where the user must first login their UNIX account where the terminal will supply the user with the personal VNC server address. Now the user must run the VNC client on his/her PC in order to connect to the UNIX environment and is required to use the server address that was provided by the terminal and then enter his password.

The UNIX desktop environment is loaded on the PC after following the above steps so the user can use all the applications he /she desire. After finishing all the tasks required using UNIX the user should quit his/her VNC session by opening the terminal and typing **vnc kill**. The VNC server will continue running in the case of only closing the client without logging out of the desktop hence the same applications will be found running the next time the user logs in to the account. The method is implemented in many universities and colleges such as in the University of Auckland's business school [155], University of Newcastle upon Tyne [156] and University of Aberdeen [157] and many more.

While the previous implementation requires from the user to download a VNC viewer to display the UNIX desktop environment, the next instance of implementation saves their users the hassle of downloading any additional software substituting that with the use of a web browser. Where this method minimises the overhead needed from the user and set any mistakes expected from the user when downloading or setting the viewer to its minimum extent however it is slower than using the standard VNC viewer. This is implemented in the University of Manitoba [158] in Canada.

Another useful implementation of VNC where it is used as an application to allow one computer to act as a terminal, viewing and optionally controlling another computer. It is employed as a useful tool for Computing & Information Services' support specialists, as it often allows them to solve a problem quickly over the phone remotely without the need to pay an office visit to the actual physical desktop. The user can also view and control his/her office computer from their home computer or from a classroom podium computer. As an example, Heriot-watt University in Edinburgh [159] uses VNC for computer support where when the PC user needs assistance from the Computer Information Support, the user calls the support line and clarify the problem his facing. The support specialist might ask to view the PC to examine it and ask the user to run the Remote Support application. Now the specialist checks up the PC and investigate any problems to resolve them. After the problem being fixed the user closes the Remote Support application, from this point further no connection is remained and no one can view the desktop until the user invokes the Remote Support application again.

VNC can thus often give you the full benefit of a support specialist's office visit without having to wait for him/her to schedule an appointment.

When you start the Remote Support application, it will ask you for the IP address of the support specialist's computer, which he will supply to you. The Remote Support application will then start the VNC connection between the two computers.

This mechanism of starting the VNC connection is based on the reverse connection method to invoke VNC. Where the Support specialist is using a listening VNC viewer and asks the vnc server on the user's PC to connect to their machine using their IP address. The support specialist will then be connected to your computer and can view what you are doing and either advice you or make any corrections from his or her own computer.

Another aspect of VNC usage is if you would like to demonstrate to a class an application running on your office computer, a VNC viewer can be used on the podium computer to access your office computer (which must be running VNC server). You can then run the application on your office computer while projecting the podium computer to the classroom. The University of Nebraska-Lincoln [160] is using VNC in Augmentative and alternative communication (AAC), which is an area of clinical practice that attempts to compensate (either temporarily or permanently) for impairment and disability individuals with severe expressive communication disorders. The staff of AAC has used VNC to provide Information Support for consultants, filed test personnel, and AAC users at remote sites. Computer programmers use VNC to troubleshoot programs at remote sites. Moreover, VNC is used in the alpha and beta testing stage of the software development where the staff installs this prototype on the consultation's computers in order to ensure that the software is working properly. Another benefit is using VNC in computer training where the staff can teach anyone in a remote site how to operate a software by downloading it to the remote computer and then using a telephone to explain the whole process.

Using VNC assisted and eased the involvement of experts placed in remote sites in the development of software during the ACC Menu Interface development stage. It brought countries together by consulting experts through North America. In addition, the University of Nebraska-Lincoln also uses VNC to support research activities and discussions, where it simplified the process and filled the gap between remote sites where a research fellow can ask for advice and feedback from his colleagues before any presentations hence edit the contents of his/her work before any major presentation.

6.1.2 Virtual Laboratories

6.1.2.1 Using VNC in hardware control

A Virtual Laboratory is a heterogeneous, distributed problem solving environment that enables a group of researchers located around the world to work together on a common set of projects. As with any other laboratory, the tools and techniques are specific to the domain of the research, but the basic infrastructure requirements are shared across disciplines.

Virtual laboratories make remote experimentation and collaboration possible when high-speed, low-latency network connectivity allows geographically separated researchers to share costly, state-of-the-art equipment without travelling to the equipment itself. Open-source tools are used to enable the researchers to collaborate on heterogeneous platforms.

One of the main goals in environments and areas of discipline that are based on laboratories is the ability to use expensive and difficult to build instruments while physically and geographically distant. This allows scientists and engineers to collaborate and access the instruments remotely. In order to achieve this goal and use many legacy applications over the Internet and make them available to remote users

“Web enabling” is the answer where it can take many forms like providing browser access to an existing application with no changes to the application's current user-interface. Or re-implement the existing application as a web-based application which has several drawbacks where it is very expensive, time consuming and it duplicates existing ‘tried and true’ software opening the door to technical problems and dual software maintenance.

Therefore using VNC to web enable legacy software and to convey the applications to the remote users is the optimal solution to provide them access to virtual laboratories without re-writing the applications or introducing instability to already stable software version.

In [161] the goal is to provide location independent laboratory where scientists could collaborate, this type of laboratory is regarded to as a *collaboratory*. For collaboratory to be effective, it may need to provide remote access to any or all of the following: instrument settings, instrument controls, notebooks, logs, analysis results, meetings and conversations. Collaboratory is coined by Wul [162] as “...’centre without walls,’ in which the nation’s researchers perform their research without regard to geographical location interacting with colleagues, accessing instrumentation, sharing data and computational resources, [and] accessing information in digital libraries” .

Virtual laboratories embrace a more general definition than collaboratories [163] where is it an electronic workspace for distance collaboration and experimentation in research or other creative activity, to generate and deliver results using distributing information and communication technologies.

VNC is used in many virtual laboratories to provide students or researchers the access to state of the art equipments. For example in a university in Sweden called Chalmers LindHolmen [164] VNC is used in the virtual laboratory in Astrophysics where they use it to control a 20 meter telescope remotely over the Internet using a regular web browser.

VNC is also used in the University of Delaware in a virtual laboratory environment [165] to share the view and control of the Mass Spectrometer where the Mass Spectrometer is an instrument which can measure the masses and relative concentrations of atoms and molecules. VNC is used in join research by faculty and students at George Washington University, Drexel University and the University of Delaware. Samples are prepared by mixing microliter amounts of matrix and analyte solutions on a probe surface. The probe is inserted into a mass spectrometer and irradiated with a pulsed laser beam. The sample probe could be prepared at the remote laboratory and shipped overnight to the instrumentation laboratory. Next day a local personal can insert the probe into the mass spectrometer and then analyse it in conjunction with personnel at the remote laboratory.

All aspects of the equipment control, data acquisition and analysis can be performed at the instrument itself or at any number of remote locations at which collaborators work. The control and analysis software is a collection of tightly coupled X windows applications supplied by the instrument's vendor and is run on an attached Sun. The matrix-assisted laser desorption/ionization time-of-flight (MALDI-TOF) mass spectrometer is controlled by application software, whose results are displayed on the computer monitor. And the Instruments internal CCD camera images are displayed on the video monitor. The X clients control the movement of the probe, start and stop the laser, adjust the high voltage settings and acquire and analyse the data, an external, secondary video monitor displays an internal CCD video camera's image of the sample in one spot on the plate.

According to the tests done in university of Delaware in theory the control of the instrument at a remote X display server is theoretically possible, however by experience not all X server worked well with all the X clients. Nevertheless, all the X clients did work well when run locally on the attached workstation.

VNC facilitated the collaboration between three geographically distant sites to concurrently share the view and control of the mass spectrometer's X desktop and share the view of a separate video stream delivered by an internal CCD camera attached to the mass spectrometer's laser apparatus as shown in Figure 6.1.

A simulation of this scenario is illustrated in video at the Internet Spring 2000 Join technical meeting on March 28, 2000.

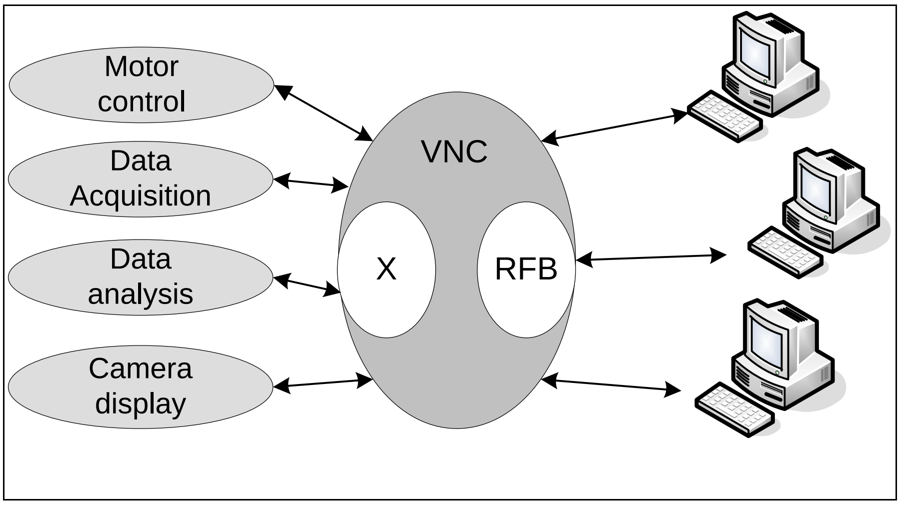


Figure 6.1 Setup for simultaneous, multi-investigator operation of a MALDI-TOF mass spectrometer [165]

The scenario is The X Windows client applications used for motor control, data acquisition, and data analysis are run on the Sun station where the VNC server is used as X server to the mass spectrometer X clients and is run on the SUN workstation that is attached to the spectrometer. The viewer is running on the sun workstation and the remote collaborator's workstations and PCs. The internal CCD camera attached to the laser is connected to an esprey-100 video capture card where Sun Microsystems's Sun Vision software sends the osprey's digital video output at 10 frames/second to a second VNC server's RFB. Collaborators local to the lab can see the camera image on a separate Sony TV monitor. The remote collaborators will start a separate VNC viewer to connect to the camera's VNC server, which uses a smaller desktop correctly sized for displaying the camera image. In addition to VNC, they used many methods to provide communication between the collaborators including Microsoft NetMeeting chat and whiteboard, SunForum's similar compatibilities and conference phones.

Another Example is in McGill University in Montréal, Canada where in [166] the paper describes a system for distance education of digital system design courses. By allowing students to take digital hardware class in which they would design, discuss, implement and debug their designs on remotely placed hardware. Field Programmable Gate Arrays (FPGA) are used and attached to the terminal to which the students have access to. This paper [166] reviewed NetMeeting and VNC as possible solutions. FPGAs require a software component for editing, compilation, simulation and programming of the FPGA and a hardware component that connects the terminal to the FPGA. The FPGA is mounted on a board that allows off-the shelf serial or parallel connection between terminal and FPGA, the Altera University board (UPI) is used. It contains 2 FPGA which are programmable via the Altera software package. There is no interface between the FPGA software and the CSCW software because the FPGA software was not designed with networking in mind therefore a software layer is used to fill the divide between the two.

For NetMeeting, Microsoft has made readily available developer studio package for creating the Interface between newly created software and the MS NetMeeting software. To provide interfacing the newly created software must have pointers to link to NetMeeting objects allowing access to the data channels between users that NetMeeting creates. Whereas VNC, does not need to Interface the Interface software with it as the FGA software can be run directly from the remote computer.

NetMeeting drawback is the increased level of complexity inherent to the introduction of third party software. Otherwise it is desirable for its videoconferencing functionality. VNC can control the remote hardware environment, which inherently allows for use of the remote hardware that is connected to the remote terminal. The problem of VNC is its lack of videoconferencing capabilities and no 'drag and drop' file transfer due to operating system differing file naming format and file structures. Therefore this paper [166] uses an implementation of both NetMeeting and VNC where VNC handles the FPGA Interfacing and NetMeeting handles the conferencing aspects of the system.

On the user's desktop, there is a VNC window, Altera software, the FPGA video and a third party software. The FPGA video for the user to view it and the third party software allows using and testing the circuit that was downloaded onto the FPGA.

According to [166] the limitations of distance laboratory for digital design are the lack of the ability to check voltage at other points than the actual probed pins throughout the FPGA. Where users cannot determine of the power supply is faulty and does not supply enough voltage to the FPGA and the user is not entirely able to detect whether it is his circuit that is not working or the FPGA board. Moreover, VNC and NetMeeting are real-time programs, they both require a lot of bandwidth yet software and hardware are ever improving.

6.1.2.2 Using VNC in software provision

In [167], VNC is used to facilitate distance learning at universities to provide students the ability to learn openGL programming. OpenGL [168] is an environment for developing portable interactive 2D and 3D graphics applications. In addition to suite of whiteboard, the only problem is performance problems due to the bandwidth intensive RFB protocol. In Figure 6.2 it illustrates the design of the virtual laboratory with the VNC server. From [167], VNC needs at least bandwidth similar to low profile ADSL connections and response times below 100 ms to work efficiently.

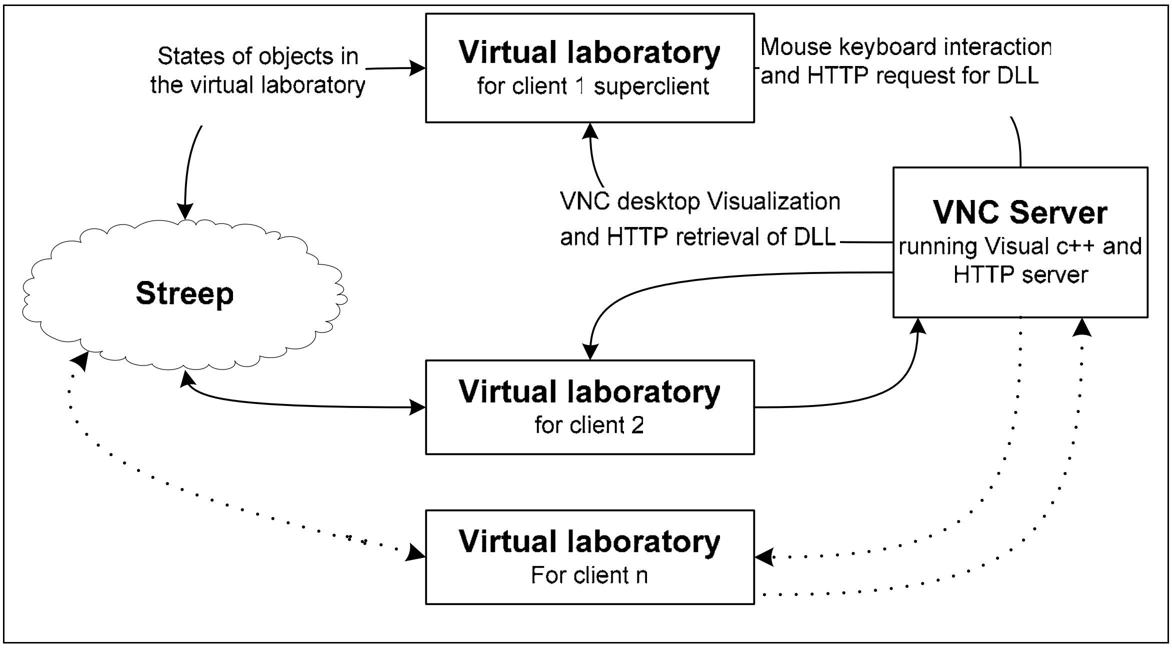

States of objects in the virtual laboratory
Virtual laboratory for client 1 superclient
Mouse keyboard interaction and HTTP request for DLL
VNC desktop Visualization and HTTP retrieval of DLL
VNC Server running Visual c++ and HTTP server
Streep
Virtual laboratory for client 2
Virtual laboratory For client n

Figure 6.2 Design and data flow in the virtual laboratory [167]

In [169] they presented a generic solution for hardware accelerated remote visualization that works transparently for all openGL based applications and openGL-based scene graphs where VNC is used for data transfer.

Virtual remote real labs offer students the freedom to conduct real practical exercises without physically attending; in [170] they provide practical exercises for Linux system installation and configuration using WebCT [171] for the course material, Rembo [172] server and client deployment and Rembo-C scripting to automate management operations, remote administration by Webmin [173] which is a web-based interface for system administration for UNIX, and VNC.

VNC has been used as a part of many projects to provide Infrastructure for tool experimentation, like the collaboration between Purdue University, North-western University and University of Wisconsin in the USA [174]. In the world of computer architecture education many Computer Aided Design (CAD) software had become a major tool to computer architects. Tools like simulators and compilers are complex and demand powerful computing resources to deliver acceptable performance level, many tools are available and shared by different courses in different universities based on infrastructure for computer architecture education to provide www portals to tools. It is based on Purdue University Network Computing HUBS and a port of the NETwork computer for computer Architecture Research and Education (NETCARE), Their University consortium consisting of Purdue University, north-western university and university of Wisconsin.

This infrastructure provides access to large pools of heterogeneous hardware resources and many other options. VNC is used as a part of NETCARE to provide remote access and control of tools of computer architecture and parallel programming with native Interactive User Interfaces. For full information of NETCARE refer to [174]. NETCARE provides universal access to tools and simulation-based experimentation in computer architecture courses.

Also, New Mexico State University's Klipsch School of Electrical and Computer Engineering implements virtual laboratory concepts by offering their students the chance to use Cadence Design Environment in research and in their coursework [175]. Cadence Design environment is an Electronic Design Automation (EDA) environment that allows integrating different applications and tools in a single framework, allowing supporting different stages of IC design and verification from a single environment.

6.1.2.3 Using VNC in Application Service Provision

Internet Service Providers (ISP) provides their services to the public to host web servers and electronic mail servers for both individuals and business organisations. The trend in the nineties has extended to also hosting applications varying from human resources modules to a computer suite and this is achieved by using the server-based computing (thin client computing) and the interaction between them and their customers is done via web browsers which are available over every PC or workstation globally. Application Service Providers (ASP) is both ISP and an application provider but not every ISP is an ASP. ASP works on the basis of 'rent-an-application' where users are not always willing to buy software they may need for a limited time so alternatively they can rent it over the Internet and also it gives the customers the ability to use applications without owning the software or the infrastructure needed to run the applications.

The advantages of ASP is similar to the advantages of a thin client architecture where instead of even centralising the data and applications in the company and dealing with complex infrastructure concerns, the company pays the ASP to be responsible of this and it removes the responsibility of any hardware upgrades and regular software updates in-house and also need not to worry about any maintenance, the only element the company has to consider or provide is a web browser, secure access through a firewall and a high speed Internet connection which can be tunnelled through a VPN. However, not running your applications on your LAN or WAN can expose you to some external factors you cannot control. Since ASP provides your application over the Internet heavy network traffic can slow down the response time and malicious attacks from hackers should be considered. In addition not every application has working web interfaces, which could eliminate its usability on in the Internet.

In [176] VNC is used in an Application Service Provision environment. Interactive Grid architecture for Application Service Providers (I-GASP) is envisioned to make computers available primarily for interactive use in a grid computing environment, where a user might access such a computer for running diverse applications such as graphical rendering, scientific visualization or mechanical CAD. I-GASP consists of a grid middleware for provisioning these computers, remote display technology that bypasses firewalls and several techniques for making the computers suitable for use in a grid environment. In this paper [176] they believe that an ASP can provide customers the access to a remote computer’s desktop for interactive use as a service. Where customers will have computers in their offices for their daily work yet will also have access to a remote computer services when computing cycles are a bottleneck and in cases when the user needs access to different processor, operating system or application suite or a more powerful computer.

In their system they address the issue of application performance. The graphic rich engineering scientific visualization and digital content creation applications are experiencing growth in their complexity in rendering on the screen which needs a high-end graphics pipeline to render such complex screens. However customers would like to access such remote applications using their normal desktops or mobile computer, therefore instead of sending raw geometric data for rendering on the client desktop, it is better to send pixels contained in the frame buffer of the allocated computer after rendering, this is accomplished by using the VNC system. This will support a heterogeneous client population. A reasonable frame rate can be sustained when an adequate network bandwidth is ensured and if a sufficient processing power for the applications is guaranteed.

VNC is used as a major part of their implementation. Where they use the VNC Proxy as a communication server which implements application level for the RFB protocol in VNC (for firewall traversal), the communication server facilitates communication between RDS and RDC by having an open port in the firewall. The remote display server RDS is the VNC server whereas the client application remote display client RDC is the VNC viewer (in their test bed).

For security, they use SSL protocol which allows VNC viewer to verify the communication encrypted using a session key shared between the viewer and communication server's identity. Only minimal changes are required from the source code of VNC proxy and viewer. VNC proxy required only one change, change the socket opened to an instance of the SSLSocket class in Java. Whereas the VNC viewer (Java version) used in their test bed had to be modified to use jsocks, a Java SOCKS client library.

6.1.3 Home appliances

The system proposed in [177] allows networked home appliances to use traditional user interface system. This user interface system supports advanced interaction devices such as PDAs and cellular phones. The prototype controls Home Appliances using PDA/Cellular phones by adopting Virtual Network Computing (VNC). Various appliances are connected to the Internet and therefore can be controlled using the appropriate device. A useful feature of the proposed system is its ability to use different devices to control the Home Appliances.

Output devices currently used in this prototype arestandard VGA screen and 3COM's PalmPilot While input devices are keyboard/mouse, NTT Docomo's i-mode cellular phone with web browser supporting compact HTML andPalmPilot
The user interface system for home appliances basically consists of Home Appliance Applications, VNC server, VNC proxy and Input and Output Devices

In this system instead of using a standard vnc viewer, a VNC proxy has been built to meet the requirements of displaying GUI of home appliances on a PDA/cellular phone. The VNC proxy conveys the bitmaps from a VNC server and converts it depending on the characteristics of the output device, and forwards input events from input device to the VNC server. As mentioned in [177] Home Appliances Applications do generate the graphical user interface for currently available home appliances to control them.

Input devices transmits an input plug-in module to the VNC proxy, this module translates input events received from the input device to mouse and keyboard events where the role of the input device is to deliver commands to control the home appliances. Output devices transmits an output plug-in module to the VNC proxy, this module converts bitmap images from the VNC server to images that can be displayed on an output device where the role of the output device is to display the graphical user interface that controls the appliances. When the system recognises the currently available appliances, accordingly it shows a graphical user interface for controlling that specific type of home appliances.
When the user wants to control home appliances using PDA, the PDA delivers an input and output plug-in module to the VNC proxy where VNC proxy changes the bitmap image delivered from VNC server to a monochrome reduced sized image using the output plug-in module. The Inputs on the touchscreen of the PDA are translated to a keyboard and mouse events using the input plug-in module and the user has the ability to transfer the graphical user interface displayed on the PDA to a larger display

by tapping on the screen and will get back to the PDA screen if the user taps again on the PDA touchscreen. The drawbacks of this method according to [177] is when controlling using the PDA the response is a little bit slow due to the overhead of JAVA in PDA.

The user can also control the home appliances using a cellular phone, where the cellular phone accesses a web server through a browser and the input plug-in module is migrated to the VNC proxy. The browser sends input events using HTTP commands to the web server where the web server forwards the input events to VNC proxy. The drawbacks of the method are, when using i-mode cellular phone to control an appliance, the output images are displayed on a PC screen. The cellular phone moves the PC cursor using its keypad numbers and it is required to adjust the cursor on the graphical user interface button for this button to be pressed and this requires some skills so [177] suggests considering more convenient scheme. This system had been useful to control HAVi home appliances (Home Audio Video Interoperability) where HAVi[5] is a digital AV networking initiative that provides a home networking software specification for seamless interoperability among home entertainment products.

Another system for using VNC in Home appliances is uVNC [178] which is a VNC server for 8-bit microcontrollers, uVNC provides small low-cost microcontrollers without graphics hardware a display screen over the network. The principle behind uVNC is to find out if VNC can be used in small low-cost embedded 8-bit microcontrollers like toasters and microwave ovens. Future applications include simple devices such as light-switches (dimmers) and thermometers in the hi-tech enabled home. With a VNC server running on these devices, each device could have a user interface that is accessible over the network. The uVNC code uses the *uIP TCP/IP stack*[6] in order to be able to communicate over the Internet. The uIP stack is intended for use in small 8-bit microcontrollers such as the intended target architectures for uVNC.

[5] http://www.havi.org
[6] http://dunkels.com/adam/uip

6.1.4 Handheld Devices

VNC has been ported to many platforms where there is a VNC viewer for Palm family (PalmVNC) [179] of hand-held computers that provides remote access to PC, UNIX or Macintosh and it is open source software. PalmVNC was originally written by Valdimir Minenko and it provides black and white access to a PC. He allowed Harakan software to take over the development of the program and they adapted it to newer models of Palm. Desktop is displayed in colour on Palm IIIc devices. It now supports server-side scaling to reduce bandwidth. It is recommended to install the PalmVNC client and WinVNC3.3.3.7 with scaling extensions. And **PalmVNC 2.0** [180] is the latest evolution of PalmVNC. It adds support for the improved screen resolutions of the latest models from Sony and Palm, and Palm OS 5 compatibility.

In addition to that there is a VNC viewer for PocketPC and PocketPC 2000 which is VNC client and it is compiled by adapting the WinCE 2.x source. This download is completely free, and is available in ARM/Xscale, MIPS, and SH3 binaries. Also Intel Xscale PXA250 (ARM) processor is supported [181]. And there is a VNC viewer for J2ME devices which is called J2ME VNC [182] which is a VNC client for J2ME devices, such as new mobile phones and the PalmOS (with J2ME runtime).

VNC has been also ported to many server platforms where you can access and control them using any VNC viewer. One of the servers implemented is a VNC server for Windows CE which currently only support uncompressed blocks. It has been ported from the original code from AT&T (Windows server release 3.3.3.r9) as part of a feasibility project [183]. In addition to that there is The Zaurus framebuffer VNC server v0.9.4 [184] which is a specialized VNC server for the Sharp Zaurus and for the Compaq iPAQ PDAs. It can be used to take control of a PDA remotely.

Also, there is a VNC server developed for web cameras called **VNCCAM** [185] which is a frameserver for AXIS and other "webcams" that speaks the VNC protocol. It was developed by Michael Rothwell. Vnccam fetches images as quickly as it can from a webcam that uses HTTP, such as the Axis webcams. It performs minimal processing, such as rotating and overlaying text, and makes the frames available via the VNC protocol and can optionally save them as jpeg files. It works with both Windows and UNIX VNC clients.

6.1.5 Mobile Phones and phones

Firstly, Looking at Mobile phones, due to the popularity of the VNC protocol and the large spread of using mobile phones, the thought of controlling your PC or workstation from your mobile was on the minds of many. And because of the thin nature of VNC that was possible without demanding any software downloads on the mobile phone, it is used to offer graphic-rich interfaces to the users. One of the free implementations of this concept is a Viewer for Nokia Communicator [186] for Geos, you can remotely access a machine running a version of the VNC server (e.g. Windows 95, 98, NT, UNIX with X-Windows, Macintosh, Acorn RISC OS) from a Nokia 9000/9000i/9110 smartphone connected to the Internet (typically via PPP dial-in).

In [187], they are proposing a VNC-based access to remote computers from cellular phones by using a cellular phone as a device for remotely controlling computers. The remote computer is assumed to be running a VNC server and attached to network.

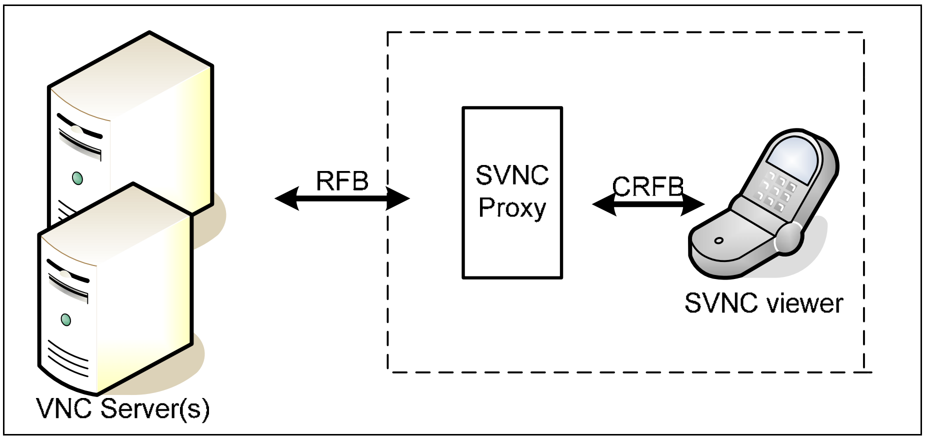


Figure 6.3 VNC-based architecture using cellular phones [187]

As it is illustrated in Figure 6.3 the VNC-based architecture consists of a VNC server in addition to Smart VNC (SVNC) proxy and SVNC viewer on the cellular phone both developed by [187].

SVNC proxy converts (crops, shrinks and resamples) 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. A proxy is placed to convert different devices, to suppress network traffic, and to support recovery from unscheduled disconnection.

The transfer is performed in their own Compact RFB (CRFB), their simplified RFB protocol. Then, the SVNC viewer displays the transferred image. Key events received by the SVNC viewer are transmitted to a SVNC proxy that converts them and sends them to the server. In order to recover quickly from an unscheduled disconnection, the SVNC proxy maintains its own database containing each session's unique information such as user name, password, target host name, other internal states. When it is first connected to a SVNC viewer, the SVNC proxy searches its own database using the user name and the target host name pair as a key. It determines whether or not there has been any previously established session. If it did exist, the proxy restores the stored state. A session's state is discarded in the proxy when the user explicitly terminates it in the viewer.

This architecture posses many benefits where The VNC protocol is an image-based protocol in which updates to a screen by applications are captured and transferred in bitmap images to the viewer. Thus, the applications running on the remote system can be manipulated by browsing the same image that we would be browsing if you were sitting at the remote PC. VNC is widely popular and available as infrastructure for controlling remote computers and for linking home appliances with PCs [177] due to the popularity of the RFB protocol.

Problems with adopting RFB in cellular phones against PDAs:

1- Cellular phone has no pointing device, so pointing is difficult. In contrast, a user can easily point anywhere on the screen of the PDAs using a stylus.

2- The screen is smaller than that of the PDAs, so it is impossible to directly show the entire display area of the remote desktop system.

The proposed architecture in [187] was implemented in Java by doing the following

1- The SVNC proxy was implemented by modifying the Java version of the VNC viewer. The proxy runs as a servlet on an HTTP server with the servlet API (Apache Tomcat 4.0) [188] as the HTTP server and installed the proxy on a PC with Windows 2000.

2- The SVNC viewer has been implemented using the J2ME wireless SDK released by NTT DoCoMo. The code of the SVNC proxy is downloaded in response to a request from the cellular phone.

The architecture has been tested and it works. To increase user friendliness and to solve the problem of the small screen, several functions are provided on the cellular viewer and frequently used screen areas of the desktop can be registered and quickly restored by using the twin view function.

6.2 VNC Prototype Environment for Collaboration and Videoconferencing

In many scenarios the ability to view multiple desktops is needed and favourable such as in the case of a supervisor in an educational organisation whose responsible of research students and have for example two students working on the same or similar project and would like to monitor their progress and be able to compare results or the case of a person responsible of his office and home PCs or workstations and while he is travelling he would like to have a simple interface to connect to both PCs and/or workstations to maintain them. Also a user such as a network administrator or someone who is in charge of many machines needs to monitor several machines at the same time he would be able to connect to them from his machine using as many viewers as he wants and he can switch between them to transfer from one screen to another. In some cases the user will want to have the sessions in front of him at the same time either to compare results or to view some applications. This Chapter starts by discussing four prototypes of providing multisessions of VNC to allow a supervisor scenario to monitor his research students. After that the possibility of increasing the productivity and affectivity of a VNC session between two participants in a VNC session by introducing videoconferencing used in parallel to VNC is discussed and a prototype is designed.

6.2.1 The development of monitoring VNC sessions' application

This section will discuss four implementations of developing an interface to provide multiple VNC sessions simultaneously and their pitfalls and the ways that they can be improved.

6.2.1.1 HTML Applications (HTAs) and Jscript prototype implementation

The main concept behind this implementation is to use the widespread and portable method of HTML and HTML applications (HTA). HTAs are full-fledged applications; these applications are trusted and display only the menus, icons, toolbars, and title information that the Web developer creates. In short, HTAs pack all the power of Internet Explorer—its object model, performance, rendering power, protocol support, and channel-download technology—without enforcing the strict security model and user interface of the browser. To create an HTA you write an HTML page and save it with the .hta extension.

Also, this section uses JavaScript and Jscript, JavaScript is a script-based programming language that supports the development of both client and server components of Web-based applications, Microsoft introduced its version of

JavaScript referred to as Jscript in internet explorer 3. Jscript is compatible with JavaScript 1.2. [189]. JScript is used to provide the built-in browser capability.

A. Vertical divided screens

The main idea here was to create a Graphical User Interface, which is divided into two parts; each part contains a VNC session. This can be used in a research team where the supervisor wants to monitor two desktops at the same time and to compare results between them. In addition to that a research student can use this GUI to monitor two machines inside of the firewall protected network. This method is based on using the browser where the user does not need to download any extra software to monitor the remote desktops. The user needs an Internet connection and the IP addresses of the remote desktops which can be supplied by the owners of the desktops.

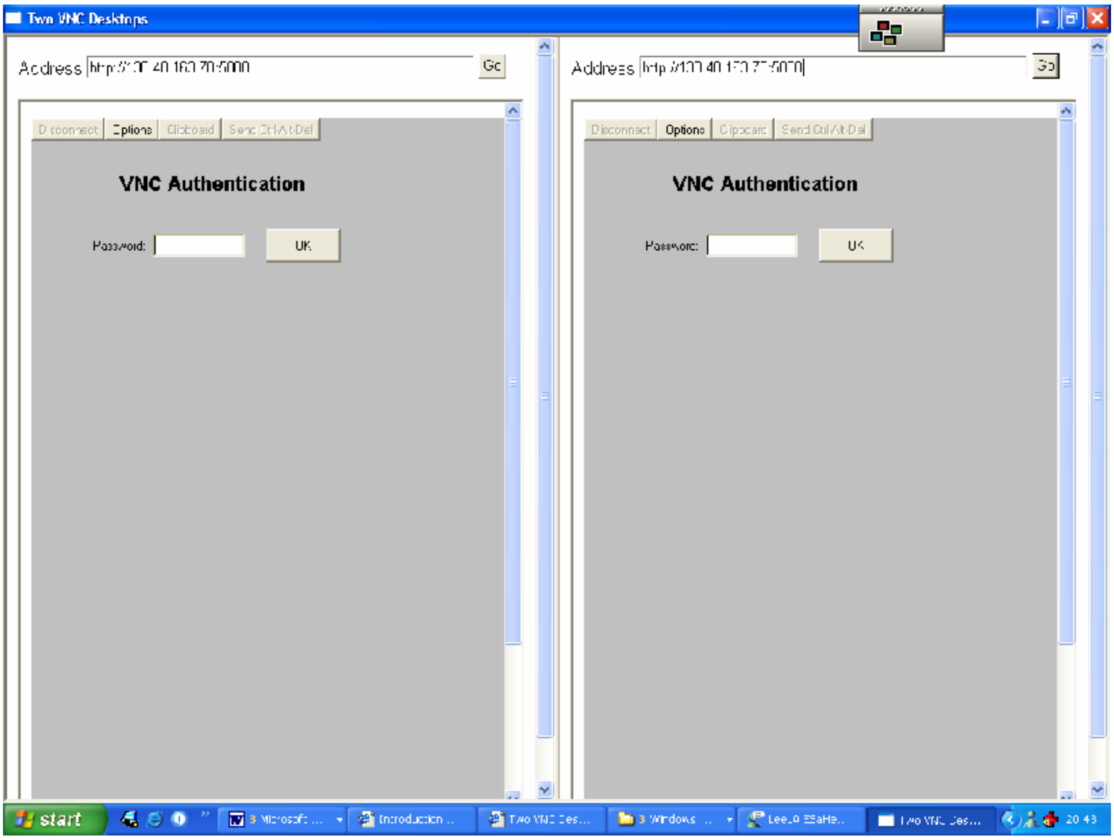


Figure 6.4 HTA vertical devided screen

In this section the steps of designing the GUI in Figure 6.4 are explained:

First, a HTML document that has an inbuilt browser capability was developed and named "browser.html" using the code in program 2 in **Appendix B**.

Second, the same code is used and renamed “browser2.html”; you can change the URL of the first site the prototype connects to in both files to suit your choice.

Third, a container GUI is developed to contain both files in a divided screen and named "VNC.hta" using the code in program 3 in **Appendix B**.

This GUI lacks maximise and minimise buttons on each part of the screen to maximise each individual session and then minimise it back to its original position. This results in having scroll bars horizontally and vertically to scroll through the VNC desktop sessions. However the middle bar between the two sessions can be moved manually right or left depending on what screen you would like to enlarge. There is no scaling support for the VNC Java version to overcome this problem.

For this method to work in a reasonable manner only two screens are used in the container GUI. I tested dividing the screen into three and four sections using the FRAMESET Tag in Program 3 but the end result was not satisfactory due to the resulting small VNC sessions in the GUI.

B. Horizontal divided screen

To overcome the drawback in the previous code where the user cannot maximise each individual screen, another implementation can be considered where various VNC sessions can be viewed in the GUI in a sequential manner and the user would be able to view each session with the whole width and height of the interface. This method also allows viewing as many VNC sessions as you like.

Figure 6.5 shows two VNC sessions. One of them is a Windows Desktop whereas the second is a UNIX workstation.

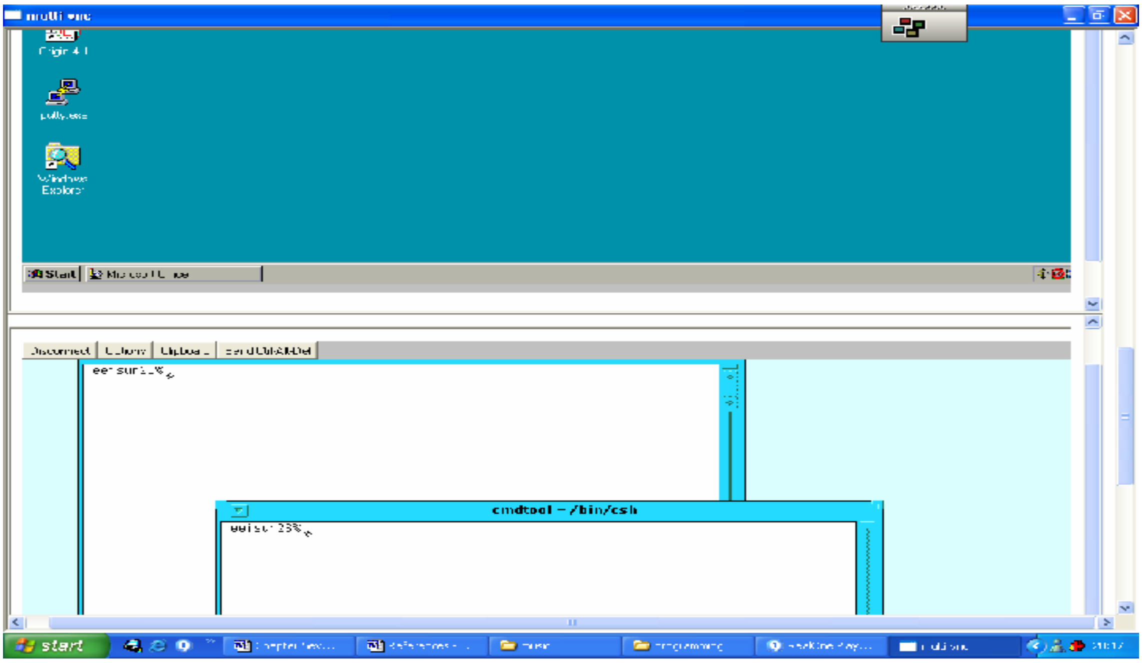


Figure 6.5 Horizontal VNC sessions

The same code in Program 2 "browser.html" in **Appendix B** is used but the container GUI is different to hold VNC sessions one after the other. You can add as many VNC sessions by including the same code of Program 2.

The main Graphical User Interface had been designed as an HTML application using the code in Program 4 in **Appendix B**.

C. Securing the HTA VNC implementation

The two prototypes can also be secured by encrypting the traffic using SSH. This can be achieved by using port forwarding explained in chapter five for both remote desktops either by tunnelling through a UNIX machine that runs a SSH server or running a SSH server on one of the Desktops. But the problem here is only the traffic between the person who is using the GUI and the machine that runs the SSH server is encrypted while the traffic between the SSH server machine and the forwarded machine is not encrypted. This is not an issue in the case of connecting to computers behind the firewall because the traffic between them is hidden from outsiders.

And in the GUI instead of typing the IP address of the remote desktops you connect to localhost. This section will explain the configuration of port forwarding on the client machine as in the example shown in Figure 5.12 in chapter five. To securely connect to both the UNIX machine and Windows machine behind the firewall, you should type in PuTTY:

Host name: ssh-server ip address

SSH->Tunnels-> **Source:** 5800

Destination: ip-address of UNIX vnc server: 5800 then **Add** and again

Source: 5900

Destination: ip-address of UNIX vnc server: 5900 and **Add**

Source: 5801

Destination: ip-address of Windows vnc server: 5801 then **Add** and again

Source: 5901

Destination: ip-address of Windows vnc server: 5901 and **Add**

And then open the SSH session and enter you account name and password. Now in the GUI such in Figure 6.4 you should type in one part: http://localhost:5800 and in the second part: http://localhost:5801 . For more information on SSH and encryption refer to chapter five.

6.2.1.2 ActiveX Prototype Implementation

ActiveX is a Microsoft technology that provides tools for linking desktop applications to www content. It enables self-contained software components to interact with a wide variety of applications. Certain components of ActiveX can be triggered by use of HTML script to provide rich web control to clients for instance, ActiveX technology allows users to view word and excel documents directly from a browser interface. MS Office applications are examples of built-in ActiveX components. ActiveX is an outgrowth of two other Microsoft technologies: OLE (Object Linking and Embedding) and COM (Component Object Model). [190]

To avoid the problems in the previous prototypes where scaling is not supported, I used an ActiveX implementation written by Thong Nguyen called VNCX [191] and previously mentioned in 6.2.2.5 that implements scaling, it is illustrated in Figure 6.6

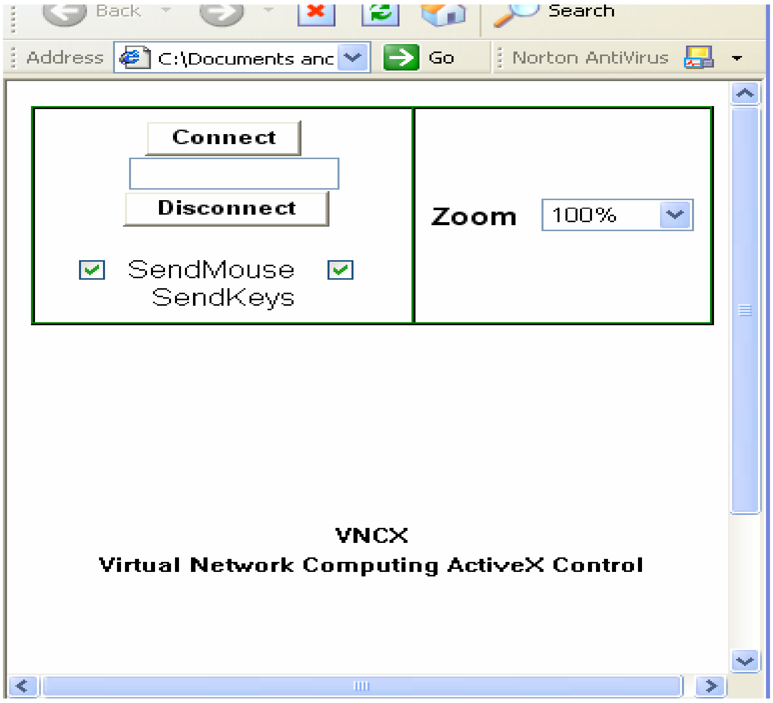


Figure 6.6 VNCX [191]

Therefore instead of using an HTML document with an in-built browser capability, I maintained the overall Interface using HTA with exchanging every frame with the VNCX ActiveX control.

The steps of designing Figure 6.7 are as follow:

Three HTML pages are downloaded from [191]; the code is in **appendix B** in programs 5, 6 and 7. Program 5 is the ActiveX control, Program 6 is the vncx_ie.htm and program 7 is vncx_password.htm they demonstrate using VNCX from a webpage.

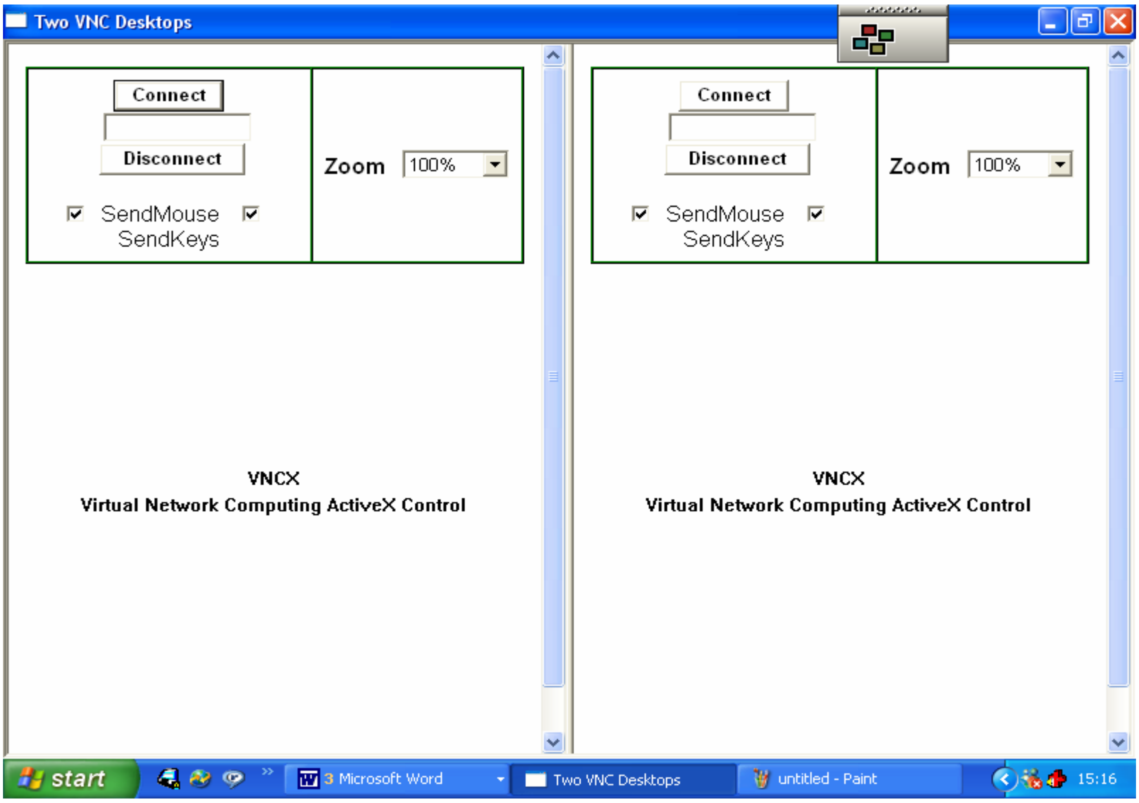


Figure 6.7 Two ActiveX Windows

These files are used in my prototype which is the GUI similar to program3, yet this GUI contains the ActiveX controls illustrated in Figure 6.8 where one frame contains a zoomed 100% VNC desktop while the other frame contains a 30% zoomed VNC desktop, the code of the GUI is in **appendix B** in program 8.

An important file is the VNCX.dll which can be obtained from [191] and must be registered in the system before any of the files can be used and to do so the user must enter the following command:

```
C:\WINNT\SYSTEM32\REGSVR32 VNCX.DLL
```

Therefore this prototype overcomes the scaling problems encountered previously.

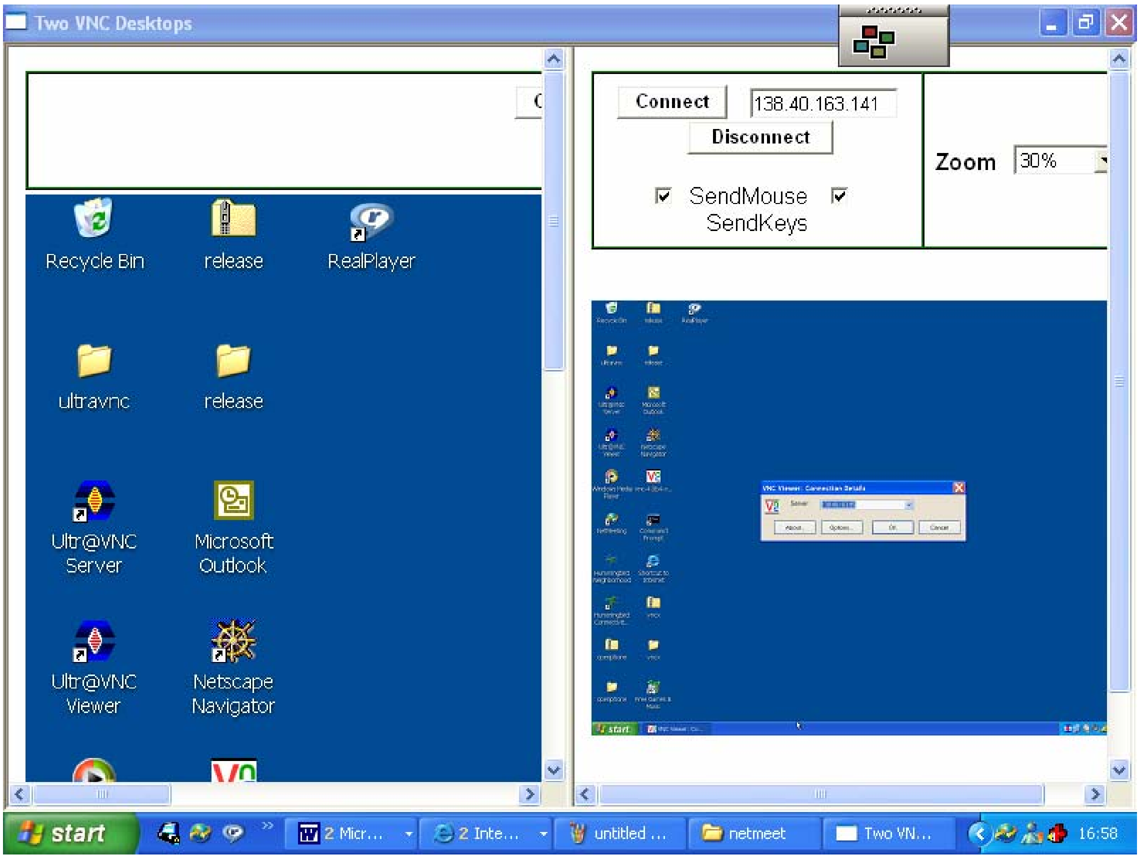


Figure 6.8 Two Connected VNC sessions

6.2.1.3 Java Prototype Implementation

The Java platform allows you to run the same Java application on lots of different kinds of computers and it provides an attractive and simple solution to the problem of distributing applications across heterogeneous network-based computing platforms.

Any Java application can easily be delivered over the Internet, or any network, without operating system or hardware platform compatibility issues. For example, you could run a Java technology based application on a PC, a Macintosh computer, a network computer, or even new technologies like Internet screen phones.

If the Java run-time platform is made available for a given hardware and software environment, an application written in Java can then execute in that environment without the need to perform any special porting work for that application. The Java virtual machine is the key to the independence of the underlying operating system and hardware; it is a platform that hides the underlying operating system from Java-powered applets and applications. And it's very easy to port the Virtual machine to a browser or another operating system [192].

The design of the Java prototype was also based on the need of having a graphical user interface that contains multiple VNC sessions. There is a main frame and the user has the choice on how many VNC sessions he/she would like to have within this frame. Figure 6.9 shows the option File->New VNC Window where the user starts a VNC Window where he/she will be prompted for Figure 6.10 and Figure 6.11

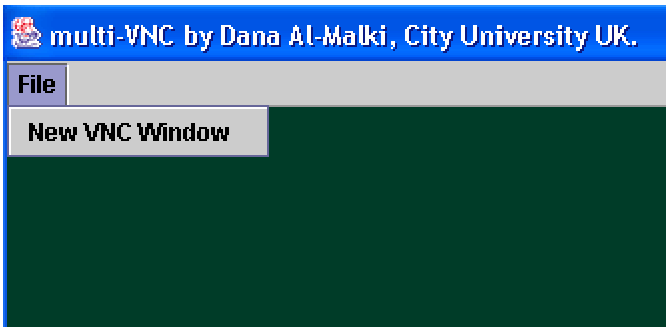


Figure 6.9 MultiVNC

Enter Host IP: where he/she should enter the IP address of the remote VNC server machine.

Enter Port: where he/she should enter the port number of a listening VNC server which is 5900-5999.

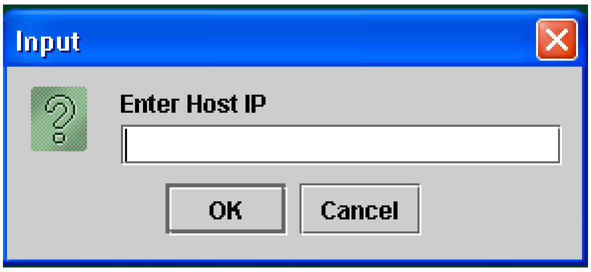


Figure 6.10 Host Name dialog box

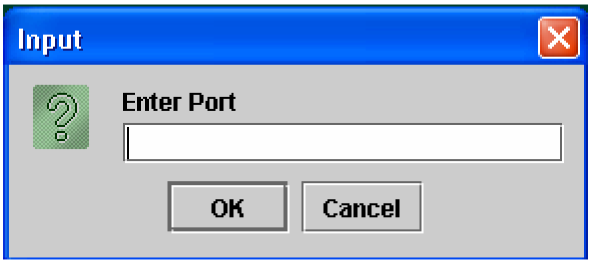


Figure 6.11 Port number Dialog box

After that the user will be shown a VNC authentication window similar to the Java VNC authentication screen. The user enters his/her password and then the remote VNC desktop will follow as in Figure 6.12

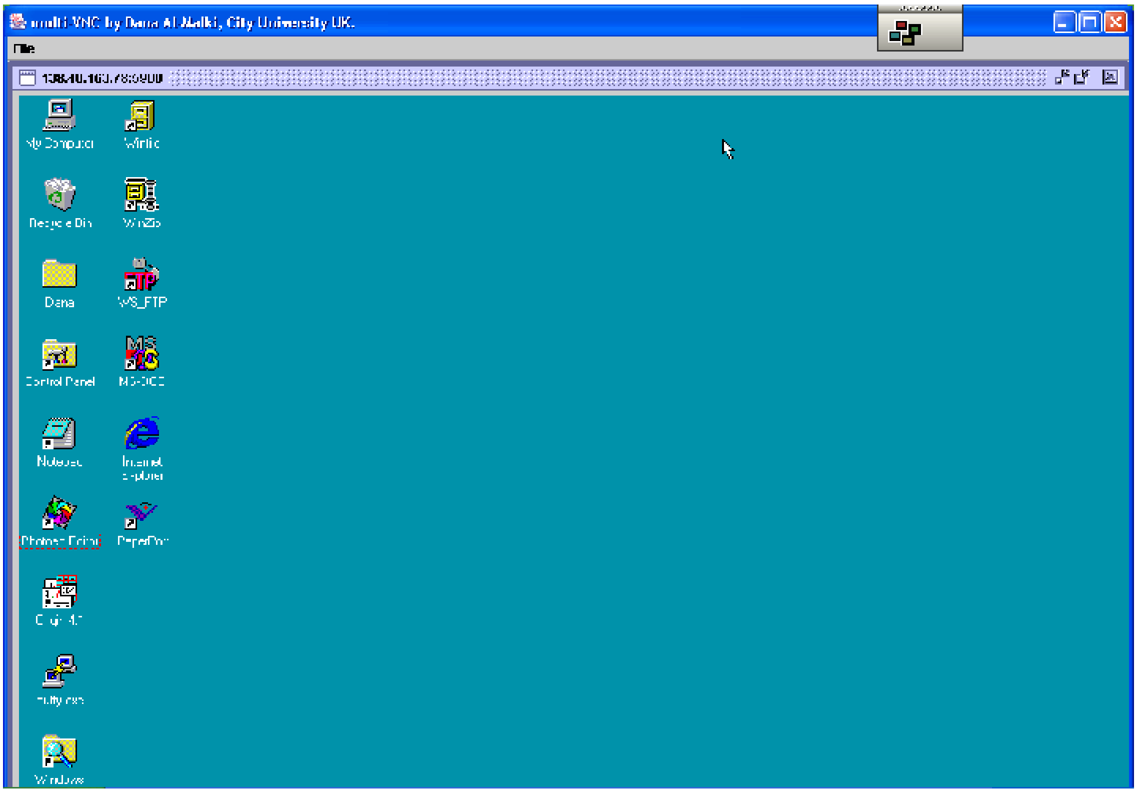


Figure 6.12 One VNC session

The user can ask for another session by using ***File-> New VNC Window*** and typing the details of the second VNC server they would like to connect to. In Figure 6.13 a Window 95 machine and a Solaris SPARC workstation are viewed.

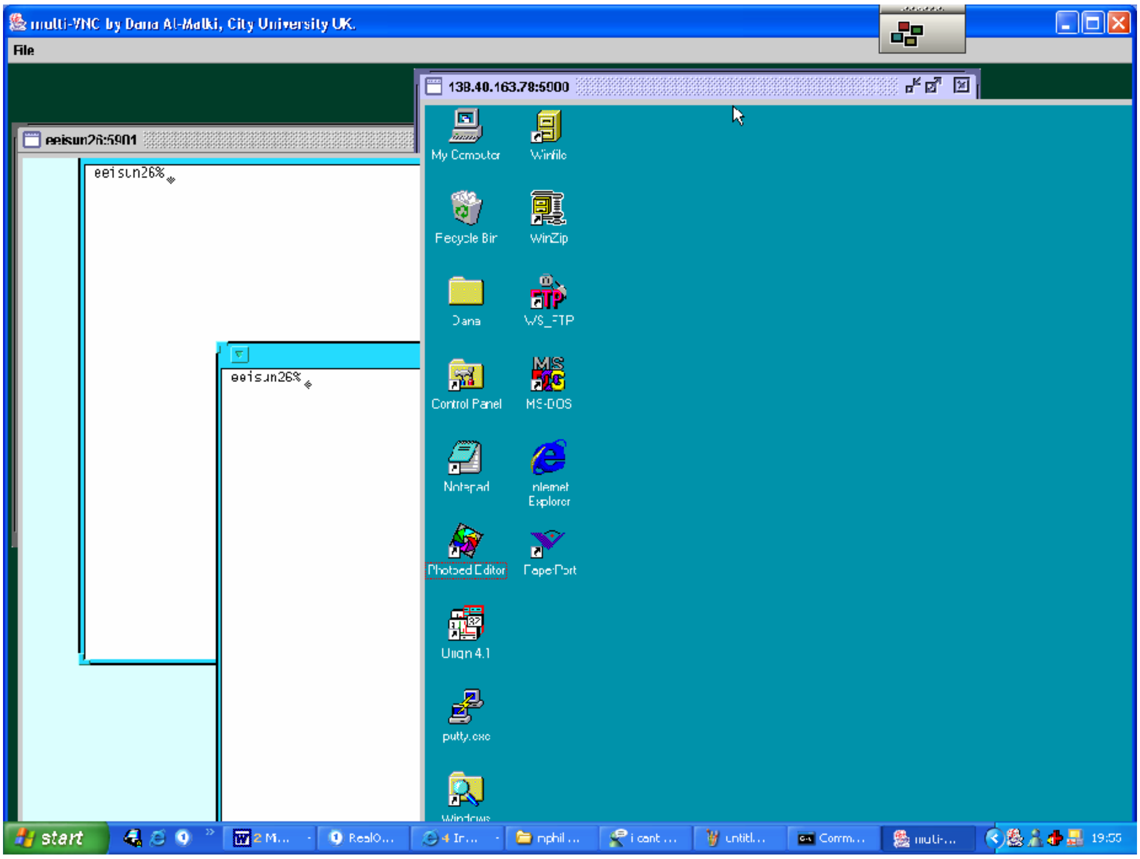


Figure 6.13 Two VNC sessions

I have developed the GUI to view VNC sessions and the code of this prototype is program 9 in **Appendix B** and named it 'multiVNC'.

In **appendix B**, the edited code of vncviewer written by RealVNC is enclosed in program 10. Only minor changes had been introduced to work with multiVNC application.

6.2.2 VNC, Audio and Video

VNC can be enhanced and enriched if another tool is introduced to enable users to make audio and video conversations. Therefore videoconferencing makes communication and collaboration richer, more effective and less expensive due to no travel costs.

VNC does not support sound where audio if played on the server side the viewer cannot hear anything therefore introducing a third party tool to achieve this property could enhance the usability of VNC in different scenarios where in our case the supervisor scenario he could ask his student to explain a part of the screen his watching, or give him his comments on his project. It also can be useful in the case of two researchers discussing a project or software running on a VNC server they both are monitoring.

As for video according to the vnc mailing list [58], VNC picks up screen data from the video card's frambuffer. The reason why video does not show on some systems, is because such video is being played into a separate buffer, which is then mixed in by the video card's RAMDAC- this is often known as "overlay mode". Overlay mode is very common on modern PC-based video cards. VNC is not capable of reading the overlay buffer and if the overlay is in use, the user will see a black hole where the video should be. For VNC to play video the video card on your machine should have a "read pixels from framebuffer" implementation and the server should not use overlay mode to display original video. And for the video to be played at an acceptable rate both the server and client should have fast CPUs to encode and decode each frame very quickly and have a high bandwidth network. Therefore it is more convenient to use a tool to provide both audio and video in parallel to VNC. Hence in the coming section I research videoconferencing and its technologies and tools to develop an understanding and to choose a suitable tool to use in parallel to VNC to provide maximum benefit between VNC users in a point-to-point session.

6.2.2.1 Videoconferencing [193]

Videoconferencing is the transmission of synchronised image (video) and speech (audio) back and forth between two or more physically separate locations, simulating an exchange as if the two (or more) participants were in the same physical conversation. This is accomplished through the use of:

1- Cameras (capturing and sending video from your local endpoint)
2- Video displays (displaying video received from remote endpoints)
3- Microphones (capturing and sending audio from your local endpoint)

4- Speakers (playing audio received from remote endpoints)

There are many examples where videoconferencing is useful like meetings, classrooms and collaboration. One of the most popular uses of videoconferencing is to facilitate attendance of meeting in situations where physical attendance of members of the meeting is hard to achieve. In this case it compensates for that and it lowers travel costs and saves time. Videoconferencing can enhance and enable meeting between people geographically distant.

The videoconferencing can be in meetings in different modes: a point-to-point call desktop mode where a single person contacts another person and starts a videoconferencing session. There is one-to-group mode where a person on the desktop contacts a group setting in a shared mode where they have on a projector in front of the group. Another mode is the group-to-group mode where the groups on both sides share the view of the other group.

Many factors affect the success of the remote participation, most importantly the quality of the audio and video. Audio is of major importance because any interruption of the audio link between the two parties can cause difficulties and make the communication worthless.

There is a multi-point meeting where more than one location is participating remotely. This case requires participants to have view of each other and to be well heard and they need a mechanism for determining who is leading the meeting and they require MCU (Multipoint Control Unit).

Another instance of using videoconferencing is in classrooms where the supervisor is monitoring his remote researchers and students in a research area where videoconferencing will facilitate their interaction. This scenario is implemented in this chapter a prototype is developed to enhance the supervisor’s interaction on a one to one basis with one of his students and videoconferencing capability is added for him to communicate with his student.

Another scenario is the introduction of a remote guest lecturer to a class where in some lectures the in-class instructor would like to consult an expert (co-instructor) in the class. Also a simple H.323 system attached to a TV or a projector allows a class to interact with a remote instructor.

A third instance of using videoconferencing is in collaboration, the videoconferencing terminal for the purpose of collaboration will come with a number of software tools including data transfer, whiteboards, ftp and chats. And an interface is often provided to enable the sharing of third party applications. Sharing tools and applications should be standardised to ensure interoperability, access and accuracy. The most common implementation is supported by ITU (International Telecommunication Union), is T.120 a family of open standards that was defined by leading data communication practitioners in the industry. Over 100 international vendors are committed to implementing T.120-based products and services. Another implementation is in this thesis where VNC is used for sharing applications and data transfer with the addition of using a videoconferencing tool.

Videoconferencing helps scientists, engineers, technologists working on a common project but geographically distant to prepare for team presentation together by application sharing and data collaboration. It is also useful when two information technology directors in two different schools are proposing a joint project in educational technologies over advanced networks, also in projects where many students are involved.

Telecommuting is another instance of using videoconferencing where it can be a link between employees who are at home and who are at the office. And to interact with colleagues of different organisations, broadband offers a less expensive solution to ISDN and it supports at least low-bandwidth H.323 calls (256-384K) holds the promise of increasing conference quality and capability in the future.

Virtual remote laboratories can benefit from using videoconferencing to ease the collaboration of researchers to share state-of-the-art expensive instruments millions of miles away. Yet it is only used for helping to maintain a virtual presence while another software is used to as an interface to control the instrument such as VNC.

A. Videoconferencing Technologies

In the area of videoconferencing the popular collaborative technologies are H.323 and Session Initiation Protocol (SIP).

1. H.323 standard

The ITU H.32x family of standards handles multimedia communication. H.320 handles communication over ISDN (Integrated Services Digital Networks) while H.324 handles communication over SCN (Switched Circuit Network) known as the traditional phone services.

H.323 communication initiated in the late 1996 and aimed at the emerging area of multimedia communication over LANs. H.323 includes voice-over-IP and IP telephony, data communication over packet-based networks like IP-based networks (the Internet). H.323 provides specification for computers, equipment and services for multimedia communication over networks that do not provide a guaranteed quality of service.

H.323 computers can carry real-time video, audio and data. The standard is based on the Internet Engineering Task Force (IETF) Real-time Protocol (RTP) and Real-time Control Protocol (RTCP), with additional protocols for call signalling and data and audio visual communications.

H.323 also specifies T.120 [194] services for data communications and conferencing in an H.323 session, T.120 support means that data handling can occur either in conjunction with H.323 audio and video or separately, the file transfer and program sharing use T.120 support to operate in conjunction with H.323 connections. H.323

products developed by multiple manufacturers can interoperate without platform limitation.

There is an implementation of H.323 which is the Open H.323 project which aims to create a full featured, interoperable, open source implementation of the ITU H.323 videoconferencing protocol that can be used without charge.

Also, there are several H.323-based clients like GnomeMeeting which is a GUI-based client for Linux written by Damien Sandras as his final year project for his degree in computer science engineering. This program allows Linux and FreeBSD users to videoconference with industry standard H.323 applications such as Microsoft NetMeeting program for Windows. Also, NetMeeting and CUseeMe are two examples of software-based H.323 clients.

It is worth mentioning that H.323 standard also defines three additional videoconferencing components to extend the functionality of videoconferencing, Gatekeepers, MCU and Gateways. The H.323 gatekeeper controls a particular set of videoconferencing resources (terminals, gateways, MCU's) and provides advanced services somewhat like a videoconferencing switchboard operator. And Multipoint Conferencing Unit (MCU) is used to connect three or more videoconferencing systems in the same conference to create “virtual meeting room”; it manages audio and video from each participant to the others such that the group communication is achieved. Also, a gateway provides transcoding services such as address translation, network protocol translation and audio/video coding translation between dissimilar conferencing technologies.

2. Session Initiation Protocol

SIP is a part of the ITEF standardisation process. SIP is a signalling protocol for establishing calls and conferences over IP networks where it is developed specifically for the Internet. It is about initiation of interactive communications sessions between users. SIP also handles termination and modifications of sessions as well. It does not define what a session is; this is described by contents carried in SIP messages. The protocol used to describe sessions is the Session Description protocol (SDP). [193]

B. Videoconferencing tools

1. **CUseeMe** -- CUseeMe [195] is videoconferencing software owned by First Virtual Communications (formerly CuSeeMe Networks and White Pine software) and created in 1993 by Cornell University. As of January 2004, many changes have taken place that had directly impacted the usage of CUSeeMe.

2. **VIdeo Conferencing tool (vic)** -- Vic [196] is a videoconferencing tool over the Internet developed by the Network Research Group at the Lawrence Berkeley National Laboratory in collaboration with the University of California, Berkeley.

Vic is based on the Draft Internet Standard Real-time Transport Protocol (RTP) developed by the IETF Audio/Video Transport working group. RTP is an application-level protocol implemented entirely within *Vic.*

Vic only provides the video portion of videoconferencing. Audio, whiteboard and session control tools are implemented as separate applications. Vat is an audio tool, wb is their whiteboard tool, sdr is a session directory tool developed by UCL. The supported systems are mainly UNIX: Linux, Solaris, AIX ¾, HP-UX9.x, SunOS, IRIX, FreeBSD, NetBSD.

3. **Robust Audio Tool (RAT)** -- RAT [197] is an open source audio conferencing and streaming application that allows users to participate in audio conference over the Internet. For a point-to-point communication only a network connection and a sound card are required. For multipoint conferencing RAT uses IP multicast capable network. RAT like Vic uses RTP standard as its transport protocol.

RAT runs on: FreeBSD, HP-UX, IRIX, LINUX, NetBSD, Solaris, SunOS and Windows 95/NT.

4. **NetMeeting** -- NetMeeting is a real time virtual conference and online collaboration tool, which enables its user to communicate with other NetMeeting participants through text, graphics, audio and video [198]. It also allows you to transfer files, share programs and operate a remote computer using a secure connection with passwords and encryption. Many Microsoft products have built-in links to NetMeeting

like Microsoft Office 2000 and Office XP and the MS Messenger service. The user can download it from Microsoft website, but it is already installed on Windows and if it is not in the menu click

start->run->conf to invoke it.

If the user is hosting a secure meeting he cannot use the audio and video features. Secure calls are intended for the transmission of data which can be protected by data encryption. Although the user can set a password with audio and video conferences, NetMeeting does not encrypt audio and video communication.

For the conversation to be clear, both parties must have properly functioning full-duplex sound cards, microphones and speakers or headphones.

NetMeeting achieves transmitting video in real-time, a person needs a video camera to send video but does not need one to receive video.

It is possible to have a picture-in-picture for a continuous two way video conversation.

Another feature of NetMeeting is sharing your desktop and programs with other NetMeeting users. It has chat feature and whiteboard and also transferring files between Windows users.

An important feature is Remote Desktop Sharing where it uses a secure connection and password to access the desktop and programs of a remote computer to virtually control the remote machine after activating the Remote Desktop Sharing the user must exit NetMeeting.

Microsoft developed NetMeeting audio and video conferencing features based on the H.323 infrastructure which allows NetMeeting to interoperate with other H.323 standard-based products. While they developed NetMeeting data conferencing features based on the T.120 infrastructure, enabling NetMeeting to interoperate with other T.120 standards-based products.

5. **DC-Share** – DC-share provides NetMeeting interoperability for UNIX users, allowing them to share applications and collaborate with users running Microsoft NetMeeting on Windows 95, Windows NT and Apple Macintosh. DC-Share is built from the same application sharing technology that is incorporated within Microsoft NetMeeting and offers multi-point application sharing and conformance with T.120 conferencing standards on Sun Solaris and HP/UX and other UNIX platforms.

SunForum Workgroup Collaboration Tools -- SunForum's Sun's [199] free H323 client for Solaris. It interoperates with NetMeeting because it includes DC-Share technology.

6.2.2.2 VNC and Videoconferencing

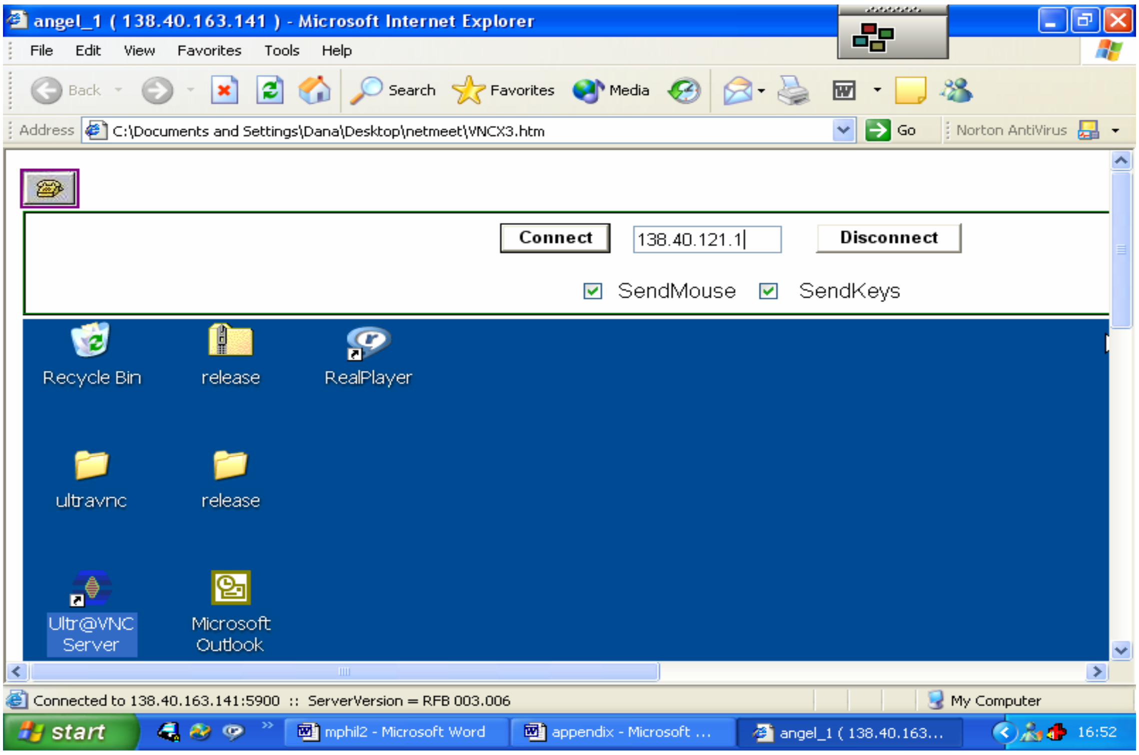
A. Videoconferencing requirements

Our environment that use videoconferencing is based on the desktop mode and it is a point to point mode where for example a supervisor is in a VNC session with one of his research students or another scenario is two research students working on the same project. And the role of the audio part of the session takes precedence over video for the reason of conveying information from and to both parties while the video is used to maintain presence therefore both parties need inexpensive camera while if any party needs the video for more than that and if it is essential a pan/tilt/zoom camera is needed to give details to both ends. For the data transfer a FTP would be beneficial. For sharing applications and desktop the VNC system is used to encourage participants to collaborate using the same application on different platforms and also because of the importance of having cross-platform system to be able to view a UNIX workstation. Therefore the videoconferencing solution we are looking for does not need necessarily to have this option; an additional benefit is having a chat facility.

Looking at our requirements I proposed using NetMeeting for my prototype which comes free on Windows operating systems, to integrate its videoconferencing capabilities with VNC because the prototype is based on Windows but also because we can extend the prototype functionality in the future and use DC-share on Solaris and GnomeMeeting on Linux to interoperate with it. NetMeeting also provides a whiteboard for sketching and a chat facility and file transfer between Windows users.

B. NetMeeting with ActiveX prototype

The ActiveX control HTML page taken from [191] is used in this prototype as a main interface where I added a button for the user to invoke NetMeeting as popup window to provide him/her with videoconferencing capabilities such in Figure 6.14.


angel_1 (138.40.163.141) - Microsoft Internet Explorer

Address C:\Documents and Settings\Dana\Desktop\netmeet\VNCX3.htm

Connect 138.40.121.1 Disconnect

☑ SendMouse ☑ SendKeys

Connected to 138.40.163.141:5900 :: ServerVersion = RFB 003.006

Figure 6.14 VNC with NetMeeting Button

The popup Window shown in Figure 6.15 is added to the Interface using the following JavaScript code added in the HEAD of the HTML page in Program 11:

```
<SCRIPT TYPE="text/javascript">
<!--
function popup (mylink, windowname)
{
if (! window.focus) return true;
var href;
if (typeof(mylink) == 'string')
  href=mylink;
else
  href=mylink.href;
window.open(href, windowname, 'width=300,height=450,scrollbars=no');
return false;
}
//-->
```

```
</SCRIPT>
```

While a link with a phone image shown in Figure 6.14 is added to the HTML page in program 11 to the NetMeeting popup window as follow:

```
<A HREF="phone.html" onClick="return popup(this, 'notes')"><img
src="phoneicon.jpg" ></A><br>
```

The popup Window is NetMeeting software embedded in an HTML page such as in program 12 in **appendix B**.

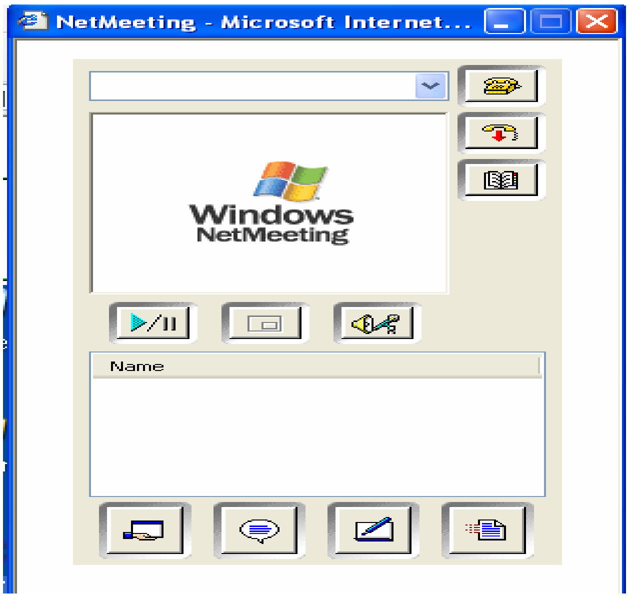
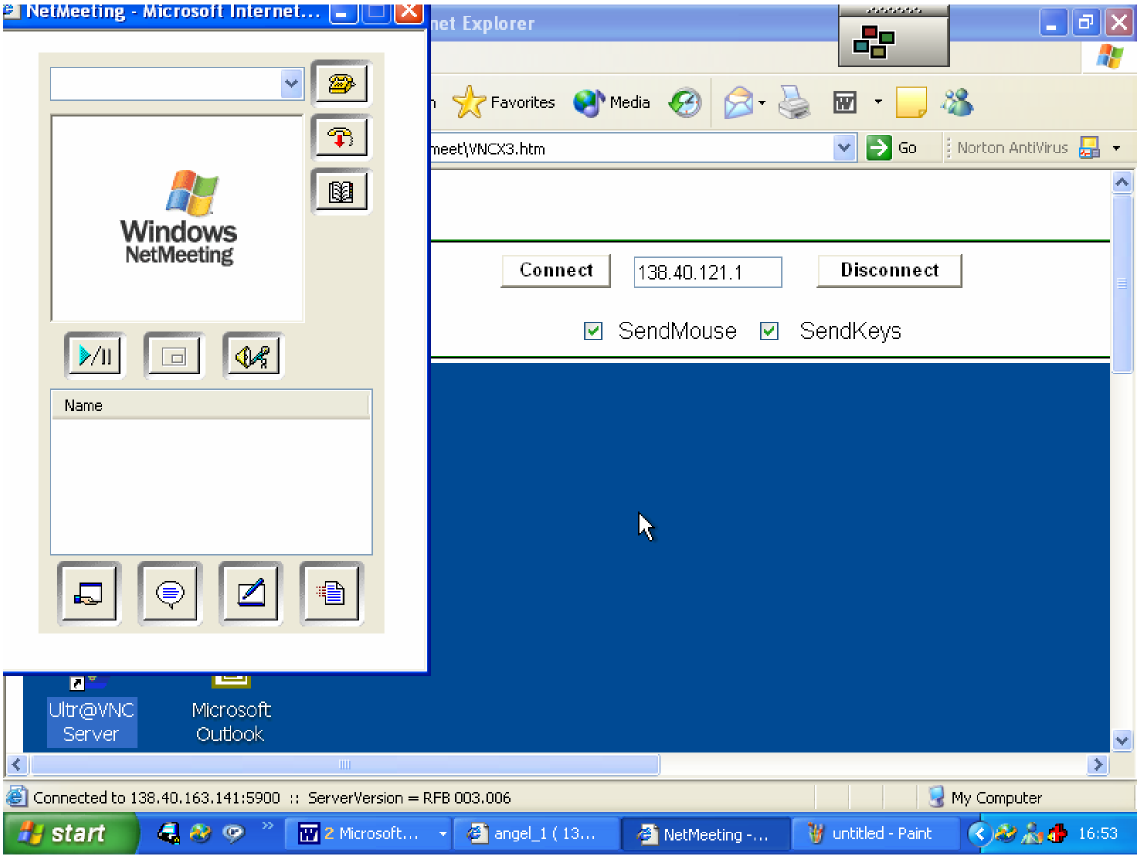


Figure 6.15 NetMeeting Window

The user must enter the already known IP address of the other party i.e. VNC server to connect in a NetMeeting session. Both parties will have audio and video capabilities to help them in their collaboration and will work in parallel with VNC such as in Figure 6.16



meet\VNCX3.htm

138.40.121.1

☑ SendMouse ☑ SendKeys

Connected to 138.40.163.141:5900 :: ServerVersion = RFB 003.006

Figure 6.16 VNC with NetMeeting

6.2.3 Conclusions

All prototypes in this chapter were designed on Windows platform but also used to view UNIX workstations.

In the first part of this subsection four prototypes which I developed were reviewed to be used in a supervision mode and were in a working state where they can be used to view few desktops to maintain, monitor and in some cases compare. The first and third prototypes based on HTML applications and VNCX ActiveX control were able to view two desktops efficiently while the second prototype also an HTML application divided the screen in a sequential manner and can view several VNC sessions in the same interface.

The fourth prototype based on Java can view several desktops up to seven was checked and due to limitation of having more desktops more was not tested.

In the second part of this subsection the idea of adding videoconferencing capabilities to VNC was discussed because videoconferencing is cost effective and saves time and effort of travelling and implemented by presenting different videoconferencing technologies and tools and then choosing NetMeeting for its functionality and the ability to expand using it in the future with UNIX workstations due to the fact that it is H.323 based and can interoperate with H.323-based tools on UNIX workstations. Both parties using videoconferencing would need an adequate bandwidth to transmit video, data and audio information and good equipment and an equivalent equipment and bandwidth at the receiving end.

It is worth mentioning that videoconferencing would not replace face-to-face meetings but videoconferencing will add value by supplementing telephone and written communication where human connectivity is required and where face-to-face meetings are not possible and are too expensive and time consuming [200].

7 Chapter Seven Improving Quality of Experience (QoE) in Collaborative Environments (CE)

7.1 Social interaction in CE

Use of computer technology often has unpleasant side effects, some of which are strong, negative emotional states. The negative emotional states can affect not only the interaction with the computer, but productivity, learning, creativity and overall well-being. Accompanying longer delays are self reports of annoyance [201], frustration, and impatience [202]. A foundation for thinking about user perceptions of web performance was suggested by Svick in [203] and it implies that the user goes through three zones through his Web experience where he is either satisfied, tolerating or frustrated. Hence these three zones start with the zone of satisfaction going through zone of tolerance ending with the zone of frustration.

For Internet users when browsing websites, the most common complaints are slow response times and difficult navigation. After waiting past a certain "attention/frustration threshold", users bail out to look for a faster site. Of course, exactly where that threshold is depends on many factors; like how compelling is the experience and if there is an effective feedback. Moreover, this attention threshold which also can be described as mentioned above as the user's zone of satisfaction is when the user is not conscious of the time it is taking to load the page. Many researches will be mentioned below to reach a rough estimation of this threshold.

A research done back in 1968 by Miller in [204] showed that users remain completely engaged in computer dialog as long as the response time is less than 1 second; Miller also found that after 10 seconds, the user's mind starts to wander and he/she loses productivity. A more recent study by Bhatti, et al. in [205] on web users buying online also show that there was a definitive negative shift of perception at 10 seconds. Interestingly, the user tolerance decreases from 10 to 4 seconds as the session progressed toward the goal of buying. Those findings are more or less compatible with Zona Research's often-quoted "eight-second rule" [206]. Zona research found that the average Web customer will wait about eight seconds for a page to download, but that current average download time across backbone connection on most web sites is almost ten seconds.

From the evidence cited above, we can conclude that the zone of satisfaction is more or less 10 seconds. That means that at 10 seconds the user becomes aware of the fact that the page is taking time to load and this is when time becomes a factor in user satisfaction. And what starts as awareness of the passage of time, slowly builds into annoyance. However, many users tolerate this page-load duration without quitting because the user's mental state or behaviour in the tolerance zone is not specifically known. Therefore, anything slower than 10 seconds needs a percent-done indicator as well as a clearly signposted way for the user to interrupt the operation. Delays of longer than 10 seconds are only acceptable during natural breaks in the user's work, for example when switching tasks. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect [207].

The user moves into the zone of frustration, when the period of time between when a user is no longer satisfied and when he becomes completely and significantly frustrated elapses. In a study by Ramsay, et al. in [208], showed users' level of interest radically changes at 41 seconds or longer where Bhatti's in [205] users indicated frustration at 39 seconds, and Selvidge in another research in [209], ran experiments that showed users experiencing significant frustration when pages loaded in 30 seconds or more.

Based on claims by Neilson in [210], saying that the fundamental usability recommendations are the same, no matter the implementation, since we are discussing user experience, not coding. It is assumed that the response time guidelines for web-based applications are the same as for all other applications.

A survey by Accenture [211] suggests that around 82% of customer defections are due to frustration over the product or service and the inability of the provider/operator to deal with this effectively. Moreover, this leads to a chain reaction as, on average, one frustrated customer will tell 13 other people about their bad experiences. An operator cannot afford to wait for customer complaints to asses the level of its service quality. Surveys have shown that for every person who calls with a problem, there are 29 others who will never call. About 90% of customers will not complain before defecting-they will simply leave once they become unsatisfied.

Moreover, when we look at the HCI community, we will find that reducing this frustration is one of its main goals. To the extent of proposing in [212] to provide computers with certain social-affective skills that humans use in helping one another alleviate frustration using a technique called "active listening" combined a careful emulation of empathy and sympathy. It is possible to say that text-only interaction, which carefully applies social-affective skills known to work in human-human interaction, can be used by a computer to provide relief of negative emotional states related to frustration, as manifest in subsequent user behaviour toward the object of the negative emotion. The result of a study in [213] based on the previous technique, suggests that designers should consider the user's emotional state as an interactive factor in the design process. Not only should designers aim to eliminate sources of frustration up front, but they should also consider building behaviours into the system to address emergent, as well as ongoing, user frustration. All of the research cited above is mainly based on human-to-computer interaction, yet we think it also can be adopted in human-to-human scenarios such collaborative environments.

Social interaction encompasses all interactivity between group members, including casual conversations and task-oriented discussions. The assumption that social interaction will occur just because the environment makes it possible and not considering the social-psychological/social dimension of social interaction that is most important in various levels of *non-task* contexts (i.e., off-task interactions); Both hold back achieving the desired Quality of Experience and social interaction in collaborative environments [214].

Therefore, we should consider that even in collaborative environments, the negative emotional states resulting from uncertainty and ambiguity in the collaborative environment can affect not only the interaction with the computer but also can cause frustration and communication breakdown between group members to the point that the frustration outruns the benefits of having the collaborative session and decreases the overall productivity. Therefore to reduce users' frustration, understanding users' requirements of having a successful collaborative session can allow us build an application that is aware of worst case scenarios and to enable them to function with acceptable quality over a network with limited performance characteristics like mobile networks.

This chapter explores how technology can be exploited to autonomously support the initiating and maintaining of social interaction, rather than, supporting participants in their efforts to establish social interaction. I would like to focus on enhancing collaborative systems with adaptable behaviour to interruptions caused by mobile infrastructure. Moreover, concentration on understanding the social dimension of a collaborative session between a number of people, can allow us to explore how applications can be designed and improved to function with acceptable quality even at situations where a network characteristics are far from ideal.

An application should be designed or improved with adaptivity in mind, in the sense that it can respond to situations where interaction and coordination are of high priority and alter the application's behaviour to suit the emergent circumstances. This can lead to increased tolerance to disruptions without reaching the point where the application's users abandon the session or where the social interaction flow breaks. Also, understanding what makes a successful collaboration can allow us to enforce application's built-in mechanism, where already available features in the existing application can be used or slightly enhanced to suit the collaborative specification.

7.2 Quality of Experience --Can it be improved and measured?

Quality of experience is a hot topic which is important in all situations where the user reaction or satisfaction is the main goal such as in interactive applications like websites, groupware, remote-control applications.

As we mentioned before, there are two Qs important to the overall user experience, these are QoS and QoE. QoS is the ability of the network to provide a service with an assured service level. Quality of Service is intrinsically a technical concept. It is measured, expressed and understood in terms of networks and network elements, which usually has little meaning to a user; it is a subset of the overall QoE scope.

Although a better network QoS in many cases will result in better QoE, fulfilling all traffic QoS parameters will not guarantee a satisfied user. An excellent throughput in one part of a network might not help if there is no coverage a short distance away. As far as measures are concerned, these statistics tell an operator very little about the level of customer satisfaction – flawless transmission of garbled packets does not make for happy users. So to infer that QoE is improved because QoS mechanisms are used to reduce jitter or average packet delivery delay may not be accurate in all circumstances.

QoE is a concept comprising all the elements of a user's perception of the network and its performance and how they meet expectations and how usable the users think the services are. In general QoE is concerned with two categories as mentioned in a white paper by Nokia [215]: reliability and Integrity which can mean different things to different applications. Reliability in this context is the availability, accessibility and maintainability of the content, the service network and/or the user device and application. Integrity on the other hand refers to the quality of the content, the bearer service and/or the software features of the user device and application.

- Examples of service availability are local and global coverage (roaming etc) included, how seamless it is for the user, what is the level of the information the service provider has given to the user and support facilities.
- Examples of Service accessibility are the success rate of user connections for any service. Service access time: the delays in setting up any service connection,
- Examples of Continuity of service are connection-interruption ratio which is the retainability of the service connection and its performance over time.
- Examples of Quality of session are application layer packet loss ratio, average bit rate achieved as ratio of bit rate demanded by application, bearer stability (bit rate variation around negotiated bit rate %), average end-to-end delay (ms or s), jitter (delay variation %).
- Ease of use: how easy is it to use service offered by the network.
- Level of support: how quick and easy is to get customer support.

Delivering high QoE depends on gaining an understanding of the factors contributing to the user's perception of the target services, and applying that knowledge to define the operating requirements. This top-down approach reduces development costs and the risks of user rejection and complaint by ensuring that the device or system will meet user requirements.

Again in the white paper by Nokia in [215] it was proposed using two approaches to measure QoE, an approach of a Network Management System using QoS parameters and Service level approach using statistical samples. The first approach is a methodology whereby hard QoS performance metrics are mapped onto user perceptible QoE performance targets.

The second approach relies on a statistical sample of the overall network users to measure the QoE for all the users in the network.

The experience is expressed in human terminology rather than metrics. An experience can be excellent, very good, fair or poor, and the operator's goal, of course, is to provide the highest quality experience to create user satisfaction and loyalty and grow their market share and revenues on this basis.

If the QoE is high, then the user is happy and satisfied. Low QoE indicates that the user does not have a good experience of the network. The concept applies to any kind of network interaction. To measure and improve QoE is a challenge that must be undertaken to assess the most accurate and complete vision of the value offered by the provider to users.

Application providers must maintain an acceptable QoE and try to improve it further to achieve customer loyalty and maintain competitive edge. A poor QoE will result in dissatisfied customers, leading to a poor application perception. User experience delay and consistency is what matters not network queuing and packet loss.

7.3 Quality of Experience in collaborative environments Framework

In light of the changing demographics of connectivity, it is important to remember that not only has technology changed, but so has user's expectations of their interaction with technology. The overall QoE for the users in collaborative environments depends on many interconnected factors both performance and non-performance related metrics, such as

- Client and server device (hardware)
- Application software that enables the user to experience the content, where the quality, ease of use and service capabilities of the groupware User Interface also have a major impact on QoE – regardless of the quality of the network infrastructure
- The network infrastructure which depends on the specific backbone(s) being used (wired or wireless), the type of bandwidth connection being used and the geographic location of the user
- End-to-end QoS such as latency and jitter as they comprise timeouts that cannot be adjusted by making an application more robust. Latency plus the transmission of that Web page to the user's end.
- other subjective factors such as user expectations, particular experience, user requirements, individual user attention span or in other words patience (very impatient, normal, very patient), cumulative frustration manifested which is the

overall time the user spent in the collaboration, Alternatives to this collaborative session, Reason of the User-experienced delay times due to: Geographic variability and number of concurrent users.

- The importance of the CE and perseverance where there are times the user will be determined to achieve the goal. User expectation in the sense a wireless connection user will psychologically adjust for the slow medium but not completely.

7.4 Quality of Experience of Disconnection in Collaborative Environments

7.4.1 Problem Scenario

In Collaborative environments, different cases would be extremely affected by the ambiguity of presence of each of its participants which can cause communication breakdown and will cause frustration to the other participants waiting for that specific user to add something to the project and they are unsure if he is coming back which will delay the communication. As mentioned before in chapter three [85] a main reason that delays impaired performance is that delay introduces uncertainty into a situation where certainty is required for expertise. When visual evidence is uncertain, fast perceptual tasks (e.g. simply watching the other person) are changed into time-consuming cognitive tasks (e.g. mentally calculating the other person's location or current activity).

In our scenario, we investigate how to improve the Quality of Experience of all users participating in a collaborative session in the case of the disconnection of one them using a nomadic connection by reducing frustration as shown below in Figure 7.1.

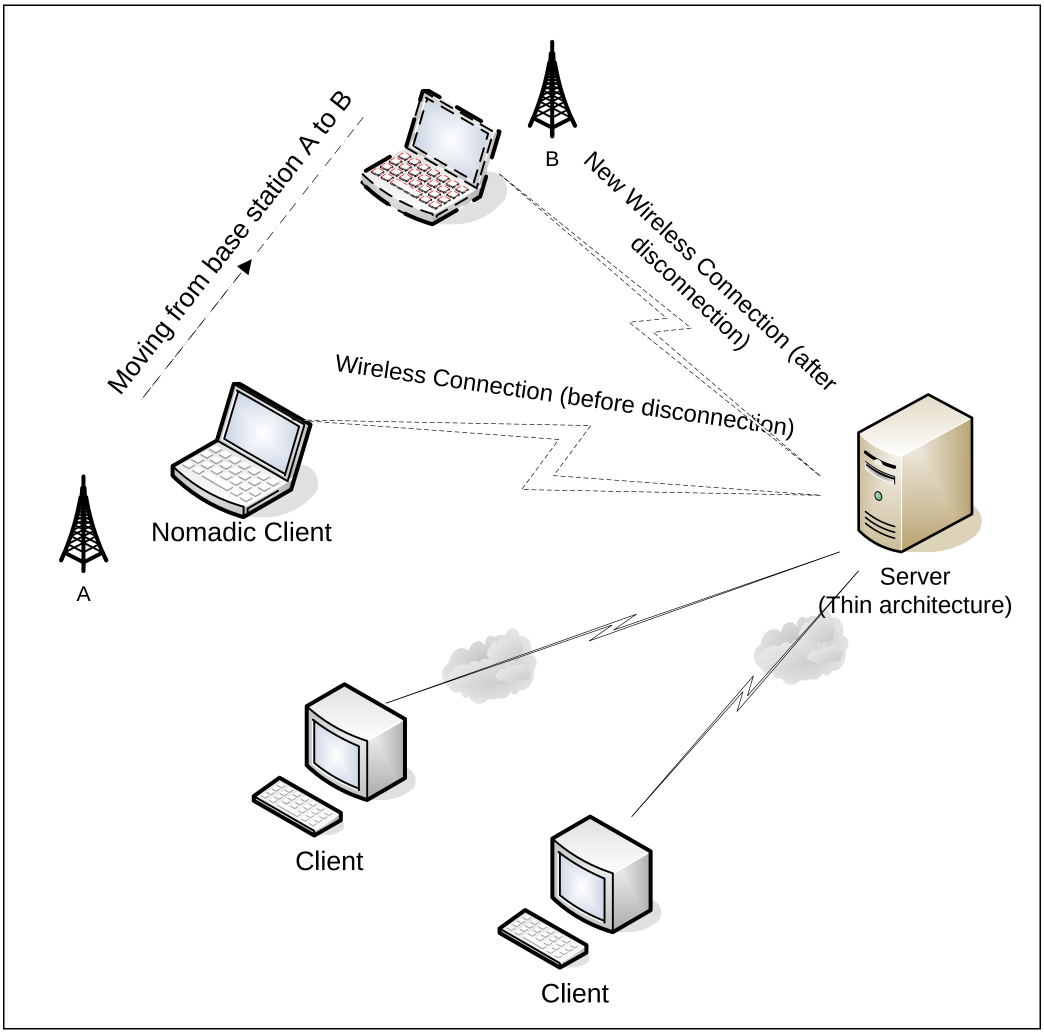

Moving from base station A to B
B
New Wireless Connection (after disconnection)
Wireless Connection (before disconnection)
Nomadic Client
A
Server
(Thin architecture)
Client
Client

Figure 7.1 Research collaborative environment scenario

7.4.1.1 The examples we focused on are

- **Research Team Meeting:** several researchers, e.g. three, are joining simultaneously to work on a common set of tightly-coupled tasks to edit, modify or build a prototype using a professional software, where the actions taken by one of them affect the sequence of the project and the result, this is an example of the importance of maintaining interaction even in the case of disconnection between group members, where it is important for all participants to be able to interact and at least be aware of the impossibility of interaction with other disconnected members.

- **Videoclip Watching:** which is a session where number of participants e.g. three connect to a server and start watching a video clip together where this is an example of the importance of consistency in the case of disconnection between group members. In the sense, the importance of efforts of different parties to reach a common static goal and users need not carry a relationship beyond the accomplishment of the task at hand such as watching a movie.

7.4.1.2 Nomadic connection

- One of the participants is using a nomadic connection while the other two using a fixed connection. In the case of disconnection such as changing the room or using a wireless connection on a portable laptop on a train (commuting).

  This nomadic node will either suddenly disconnect or will need to perform a handover where the goal is to associate with the access point that provides the best link layer connection at time and that can be used for extended period of time.

7.4.1.3 Assumptions

- Therefore, we are assuming that the mobile node is using signal quality measures to predict handover before disconnection. These quality parameters are usually provided by the driver of the wireless LAN hardware and can be gathered easily. This leads to the main idea to continuously monitor the signal quality and, upon exceeding or falling below certain thresholds, alert concerned applications. With these abilities a mobile node can control the handover process and perform fast handovers. By taking over control, a mobile node is also responsible to conduct the handover by itself. One of the handoff algorithm mentioned in chapter 2, is Relative Signal Strength with Hysteresis and Threshold (RSSHT).

7.4.2 Analysis: Characteristics that affect QoE of a disconnection

7.4.2.1 To Wait or Not to Wait, that is the Question? (Disconnection Awareness)

In general when one studies Awareness, there are several elements of workspace awareness such as those mentioned in [71] some of which are, presence (who is participating in the activity?), location (where are they working?), actions (what are they currently doing?), extents (what can they see? How far can they reach?), expectations (what do they need me to do next?).

Therefore, what matters in the scenario we are investigating is a specific type of awareness related to feedback and disconnections. This is disconnection awareness which enables all participants to know if one of them, e.g. a mobile node, has disconnected due to handover or a weak signal.

Previous research by [207] suggests that providing continuous feedback reassures users that the system is working and gives them something to look at while waiting. Therefore, this finding suggests that users require feedback to be assured that the network is continuing to process their request. Results in [216] show that users' conceptual models of the way in which networks operate can significantly influence their tolerance of QoS in predictable ways. Consequently, an understanding of users' conceptual models, and, perhaps more importantly, the behaviour which is driven by them, is a crucial step in accommodating user demand.

In this case it is human-to-human interaction rather than the human-to-computer interaction mentioned above because in collaborative environments it is important to have feedback between all members.

The knowledge of the presence of delay prompts the user to adopt a different coping strategy than he would have otherwise used. Also, knowing about the presence of delays avoids the problem of the user incorrectly dividing responsibility for failures to the system, the interface or his own skills [216]. Knowing the latency users will tolerate also allows servers to provide the appropriate feedback to the clients. Therefore providing information concerning the disconnection of a client can significantly increase the remaining users' tolerance of poor QoS. For example, the nomadic user can initiate informing the server of the disconnection after examining signal quality where the server receives this message and then it will be possible to process the message as GUI screen informing the other clients of expecting no feedback from that client until he returns to the session and this can cause that the QoE of the task as a whole is seen as better. So the disconnection awareness mechanism is receiver-driven, since the receiver is best able to determine the QoS that it is experiencing.

Many questions should be answered after a client disconnected and before notification,

- What is the appropriate threshold that the server should not exceed before notifying the rest of the participants of the disconnection?
- After the notification, the question is how long the rest of the participants could wait for the disconnected member to return? What is the threshold?

The importance of informing all clients in an ongoing collaborative session where there is a closely-coupled interaction between few users working on a project, is that the loss of getting any feedback will introduce many delays in the form of others wondering where the participant is. In case if the participant was the expert, the communication session would breakdown because of the critical role he plays in the collaboration session and the frustration from the other two clients not knowing if he is going to return or not, and the productivity of the collaborative session will be reduced to the point of maybe ending the session and not continuing the collaboration anymore.

7.4.2.2 Client's Session disconnection Amnesia (Session Consistency)

The virtual environment is usually partially or totally replicated on each participant's host, to reduce the amount of communication required; using this technique, only the updates or changes of replicated state need to be sent to the participants. Because of network and processing delays, the state updates take some

time to reach the destination and they can take different amounts of time for different participants. The resulting inconsistencies between state replicas created by network delay on local area networks are usually short lived and pass unnoticed to the users.
The deployment of Collaborative Environments over wide area networks results in much larger network delays, with correspondingly larger inconsistencies between the replicated versions of an environment. The network delay caused by network distance, geographical distance and low-bandwidth communication (e.g. from dial-up links) is likely to be a major factor hindering the development of the globally distributed interactive shared virtual environments of the next decade.

In the research team meeting and videoclip watching scenarios mentioned in section 7.1, we are specifically concerned with the scenario of several users watching a movie together, and we refer to consistency in a different context where it is important for all the users to have a uniform video transfer of the movie in case if one of them disconnected, due to its wireless connection's weak signal or the process of handover, and then reconnected. That means that one of the users will lose video frames or session's data in the period it had disconnected and because the other users are still watching the movie, he can miss a crucial part on which the plot of the movie depends. Therefore to overcome this drawback of losing a part of the movie is to freeze the transmission on all links until the disconnected client returns.

When the server realises that a disconnection happened it should perform one of two actions

1. freeze session on all links (watching video)
2. disable input and output (collaborative session)

Until the disconnected node comes back (re-joins the session) and then it switches to normal mode

IF disconnection happens **THEN** server freeze session **UNTIL** disconnected node returns **THEN** session is unfreezed

This comes at the expense of adding an additional delay component to the end-to-end latency. Hence, an application in a collaborative environment can add an x delay on all links to preserve the quality of collaboration when it detects a disconnection of a moving mobile client.

We have to find the answers of many questions such as:

- When does the freezing start?
- What happens to the lost packets meant to the mobile node (video frames) in the time that the MN disconnects and the server realizes it is disconnected?
- Is using a MN-initiated disconnection will decrease lost packets?
- How can we measure the loss?

7.4.3 Framework: Improved Collaborative Quality of Experience (IC-QoE) in disconnections Algorithm

7.4.3.1 IC-QoE Algorithm

This section will detail the steps in reducing frustration in the scenario explained in 7.1 by explaining Improved Collaborative-QoE (IC-QoE) algorithm, which is the proposed mechanism to increase tolerance to disconnections when applied to collaborative systems. In other words, Improved Collaborative Quality of Experience (IC-QoE) algorithm is the QoE model for disconnections in collaborative environments and was developed to **equally** improve the Quality of Experience for all users of a collaborative session in the case of one of them disconnecting and can be extended to cater more than one disconnection.

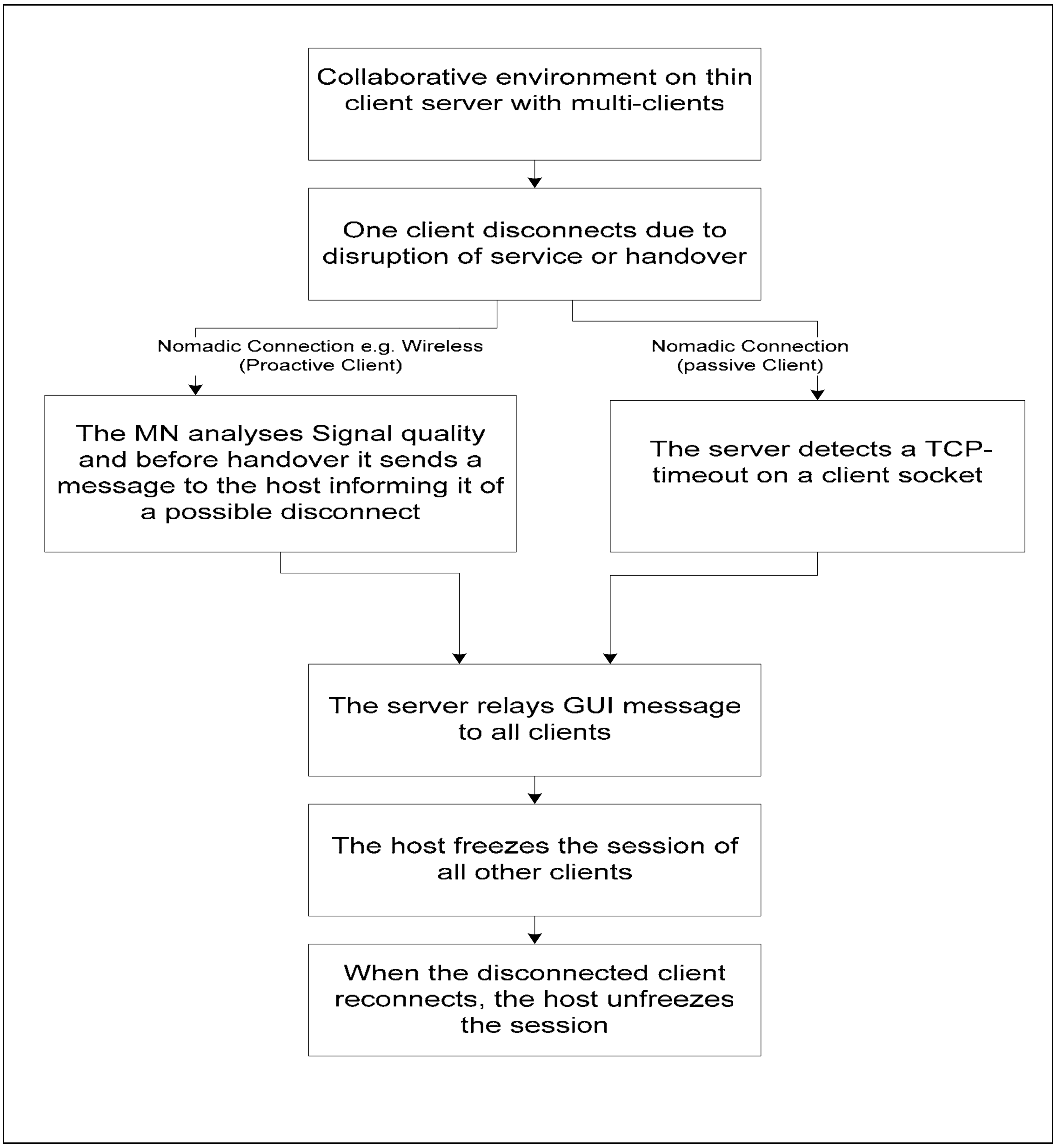

Collaborative environment on thin client server with multi-clients
One client disconnects due to disruption of service or handover
Nomadic Connection e.g. Wireless (Proactive Client)
Nomadic Connection (passive Client)
The MN analyses Signal quality and before handover it sends a message to the host informing it of a possible disconnect
The server detects a TCP-timeout on a client socket
The server relays GUI message to all clients
The host freezes the session of all other clients
When the disconnected client reconnects, the host unfreezes the session

Figure 7.2 General (IC-QoE) Algorithm

1. The concept of the IC-QoE Algorithm is based on a Collaborative environment with multiple clients. For example, Researchers or professionals geographically dispersed around the world frequently need the opinion of other experts in different continents and it is not feasible or possible to travel because of time or cost restrains, thus it is more convenient to use a collaborative environment or in other words a groupware system and specifically we are interested in thin client systems to facilitate their work, Also several users may start watching a movie on a distant server. One of the clients is using a nomadic connection

either by using a laptop with a wireless connection while travelling or someone who is on a business trip using a hotel room's connection and is changing the room for some reason or using the lounge connection instead.

2. The algorithm addresses two approaches based on the client's capability to inform the server of its disconnection due to disruption of service or handover.
    a. The first is the proactive client where it is the disconnected client's job to inform the server of its impending disconnection by means of analysing Signal-to-Noise Ratio and before handover it sends a message to the server informing it of a possible disconnection.
    b. The second is the passive client where it is a more general method that depends on the server detecting a TCP-Timeout on a client socket.

3. The server when acknowledged of the disconnection (by proactive or passive client approach) should relay GUI messages (using thin client characteristics) to all clients satisfying the Disconnection Awareness characteristic.

4. Then the server if possible should freeze the session of all currently connected clients to satisfy the Session Consistency characteristic by intending to have a uniform session to all users throughout disconnection and therefore maintaining a better Quality of Experience to all users equally.

5. Then when the disconnected client reconnects the server should unfreeze the session to all the other clients.

7.4.3.2 IC-QoE parameters

We define collaborative performance in terms of the receiving user's Quality of Experience (QoE). The QoE model represents characteristics that affect the end user's ability to easily find out the disconnection of one member and how long will the system take before informing them of that and how closely similar the session is on all links even in the case of the disconnection (before and after). These characteristics are Disconnection Awareness and Session Consistency. The overall components are shown in Figure 7.3.

Optimal QoE would have participants aware of the disconnection of a node because of the handover and having approximately similar sessions. For our case, before and after disconnection

There are several factors that affect the choice of the approximate values for these parameters, including the expected group organization, the task, and the group size.

At the network level, QoE must be translated into factors that can be measured and manipulated. We use three typical QoS parameters: latency, message loss rate (also called message error rate), and jitter. Since the receiver may never get the source data in its entirety, due to loss, we cannot directly measure disconnection consistency in comparison to the source stream. Also, the disconnection awareness can depend on both the latency and Jitter parameters. Latency between the time the MN disconnected and the time, all members of the collaborative session, have been informed of this disconnection (receiving the GUI). And Jitter affects whether the other members will think that if the node's telepointer has stopped due to him finishing action or being disconnected.

A. **Disconnection Awareness** considers whether the participants are aware of the disconnection of one of them (e.g. Mobile node due to handover). A proposed solution before disconnection the mobile node sends a message to the server to inform all participants of its temporary disconnection (must decide what is the maximum time the participants should wait, which can be the average handover delay of 6-12 seconds [217] or the satisfaction threshold of 10 seconds. People engaged in a closely-couples work are likely to have much more interest in the details of the people working with them.

Therefore, we conclude that disconnection awareness is important in a collaborative environment for shared workspace and to govern accessing shared resources. Therefore, to answer the previous questions in 7.3.2.1 it will be useful if the application can tell if a client is disconnected for more than 10 seconds, and if not then process the action of displaying GUI to inform users of a disconnection where the server can employ a timeout mechanism checking on the clients' sockets, that could warn the user when a reasonable amount of time had passed and the re-connection had not happened.

A prototype of the awareness mechanism GUI is mentioned in detail later in 7.3.5, where the source code is written using the Java language for reasons explained before.

B. Session Consistency considers whether the mobile node-in the case of a disconnection- have the same view of the shared workspace session or video clip compared to all other participants before and after the disconnection. One of the proposed solutions is to calculate the session packet loss for the mobile node and compare it roughly to the session packet loss of all other participants.

The issues the designer must consider is how to make it possible for the algorithm to decide the amount of time that all other users have to wait before pausing/freezing the session in the case if the mobile client has no means to know when is it going to disconnect. We can build the assumption on the attention "satisfaction" threshold and frustration threshold therefore the freezing starts directly after the awareness screen is displayed i.e. 10 seconds. Also we can consider that 40 seconds is the maximum of freezing the session because according to evidence cited in section 7.1, 40 seconds is the frustration threshold.

When assessing the requirements of the consistency mechanism, and after researching different methods of improving TCP in mobile environments and looking closely at some of the methods already developed as well as mentioned in chapter 2 in 2.5. We chose the freeze-tcp because it requires only the mobile host's TCP code to be modified without any modification made to the intermediary i.e. to the base station or the sender and this is mentioned in more detail in 7.3.5 and in Appendix D. Another reason is because freeze-tcp satisfies several points of the requirements of our IC-QoE Algorithm where freeze-tcp is a proactive mechanism in which the receiver notifies the sender of any impending `blackout' situation (e.g., due to wireless channel fading or handover) by `zero window advertisement (ZWA)' and prevents the sender from entering into congestion avoidance phase. Upon receiving ZWA, the sender enters into the `zero window probes (ZWP)' mode and freezes corresponding timers. While in ZWP mode, the sender transmits zero window probes and the interval between successive probes grows exponentially until it reaches 1 minute, where it remains constant.

When the `blackout' period is over, the receiver sends TR-ACKs (Triplicate Reconnection ACKs) for the last data segment successfully received to enable *fast retransmit* at the TCP sender.

Ideally, the `warning period' prior to disconnection (i.e., how much in advance should the receiver start ZWA) should be long enough to ensure that exactly one ZWA gets across the sender. If the warning period is longer than this, the sender will be forced into ZWP mode prematurely resulting in idle time prior to disconnection. If the warning period is too small, the receiver might not have enough time to send out a ZWA which will cause the sender's congestion window to drop.

The advantage of this approach is that it offers a way to resume TCP connections even after longer interruptions of the connection. It is independent of any other TCP mechanism, such as acknowledgments or sequence numbers, so it can be used together with encrypted data. Also, the advantage of Freeze-TCP is that it avoids slow start congestion during handoff, whereas with standard TCP goes into slow start when it detects a disconnection during the transfer. With this enhancement, TCP throughput performance with handoffs should greatly improve.

However, this scheme has some disadvantages, not only the software on the mobile host has to be changed, to be more effective the correspondent host cannot remain unchanged. Also, the receiver needs to predict the impending disconnections. For this, some cross-layer information exchanges may be necessary. Therefore all mechanisms rely on the capability of the MAC layer to detect future interruptions.

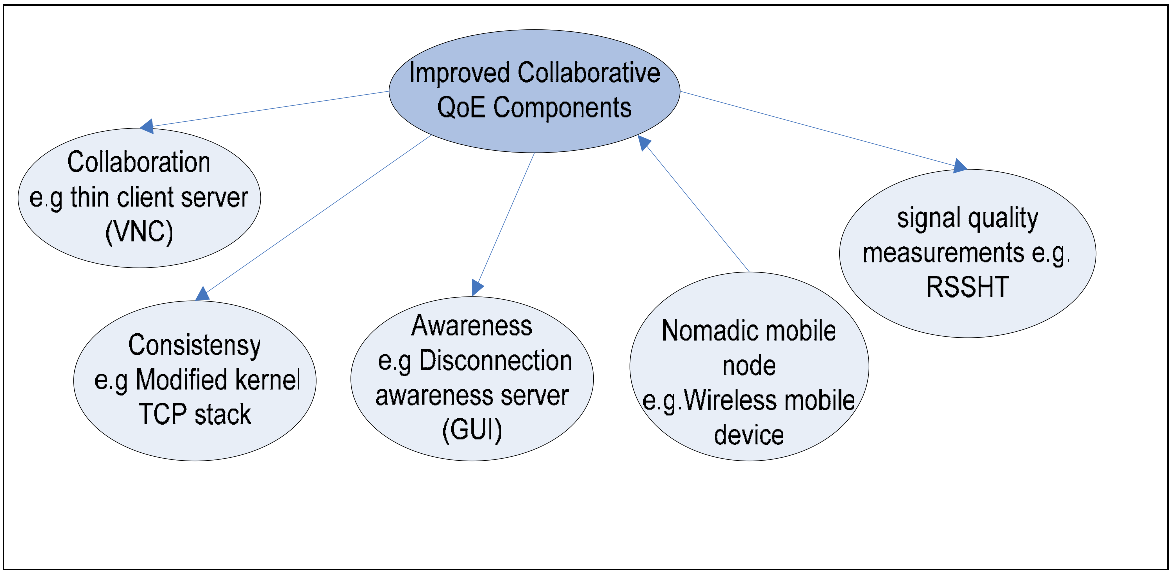

Improved Collaborative QoE Components
Collaboration e.g thin client server (VNC)
Consistensy e.g Modified kernel TCP stack
Awareness e.g Disconnection awareness server (GUI)
Nomadic mobile node e.g.Wireless mobile device
signal quality measurements e.g. RSSHT

Figure 7.3 Improved Collaborative-Quality of Experience components

7.4.4 Design Approaches of IC-QoE Algorithm

As mentioned before, the algorithm addresses two approaches based on the client's capability to inform the server of its disconnection due to disruption of service or handover. We will mention in this section in more detail both the Proactive client approach and the passive client approach.

7.4.4.1 The concept of the Proactive Client

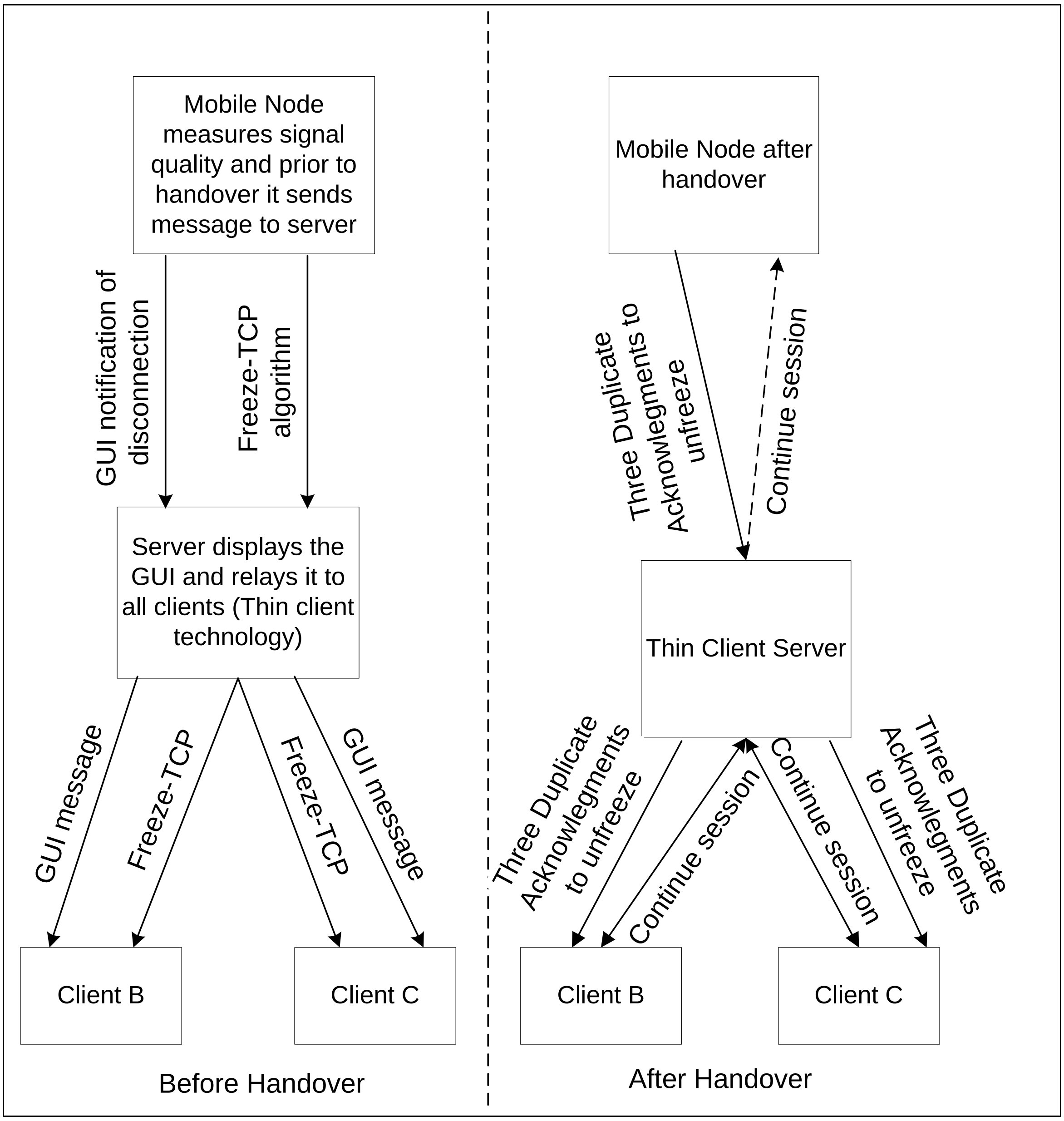

Mobile Node measures signal quality and prior to handover it sends message to server
GUI notification of disconnection
Freeze-TCP algorithm
Server displays the GUI and relays it to all clients (Thin client technology)
GUI message
Freeze-TCP
Freeze-TCP
GUI message
Client B
Client C
Before Handover
Mobile Node after handover
Three Duplicate Acknowlegments to unfreeze
Continue session
Thin Client Server
Three Duplicate Acknowlegments to unfreeze
Continue session
Continue session
Three Duplicate Acknowlegments to unfreeze
Client B
Client C
After Handover

Figure 7.4 Proactive Client Approach

In this mechanism the main issue is how would the server know that the disconnected client is going to return, we assume that client initiated the disconnection, notifying them of the need to disconnect from one room and connect from another; presumably the disconnection time would not take more than few minutes. If a client disconnected without the other members of the collaborative environment knowing, It can result in communication breakdown and user frustration where they are expecting the member to be present in the session and waiting for his feedback and what they get is no feedback and predicting if he is online or offline which will tear the dynamics of the session and the quality of the whole experience.

In Figure 7.4 the operation of the proactive client approach in the IC-QoE is shown in detail based on the concept of Freeze-TCP algorithm [43]. Its operation is divided into two phases, before and after handover. Before handover, where as mentioned before the handover is mobile-initiated using handover initiation algorithm based on relative signal strength measurements such as hysteresis and threshold. Therefore before it performs the handover it sends a message to the server informing it of its expected disconnection due to handover and using Freeze-TCP algorithm it advertises a zero window to the server so the server will freeze the session and then in accordance the server relays the GUI screen to all users to inform them of the disconnection and then it uses Freeze-TCP to freeze all other sessions.

After handover, the mobile node sends three duplicate acknowledgments to the server for it to unfreeze the session, and the server respectively sends to all other clients the three Duplicate acknowledgements to continue the session.

7.4.4.2 The concept of Passive Client

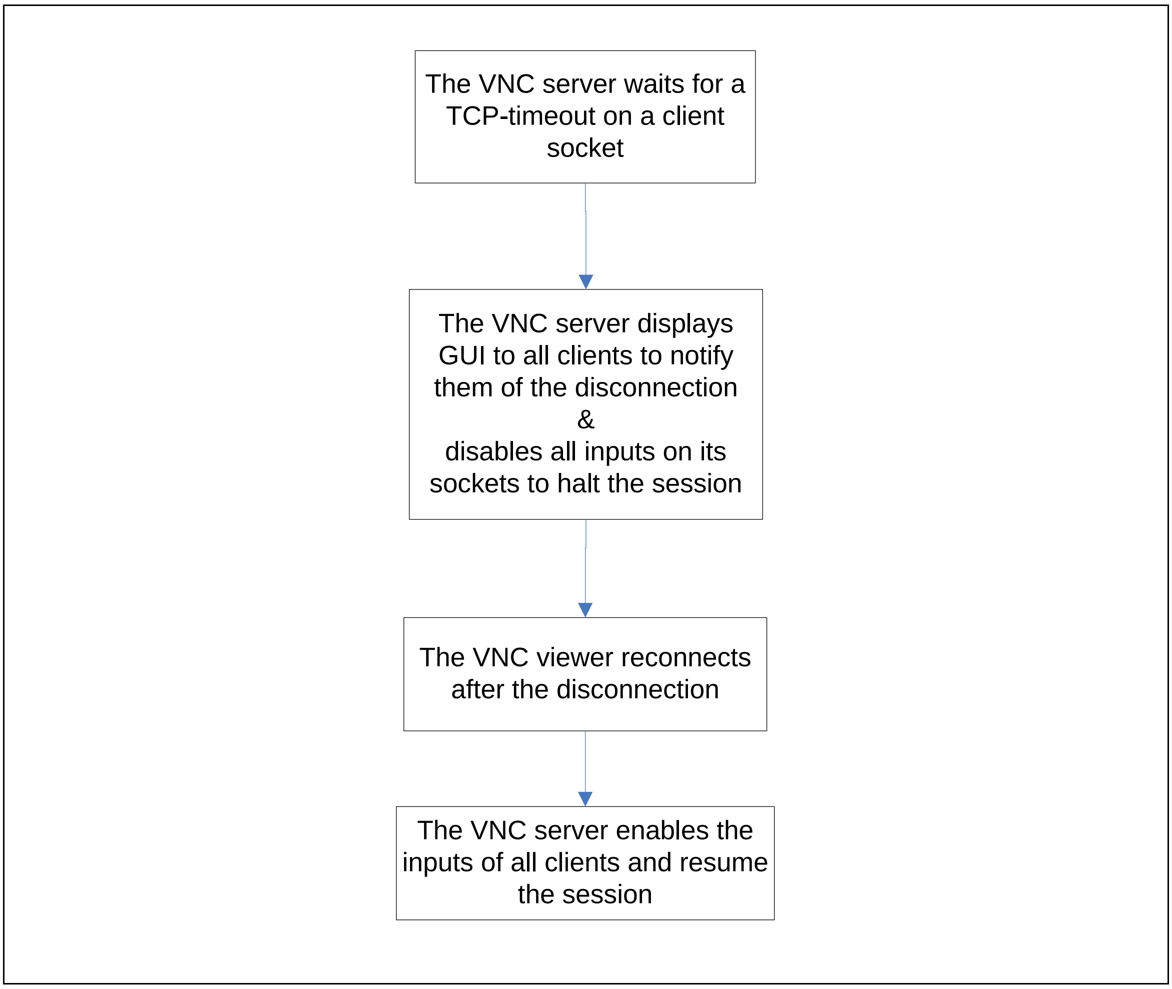

The VNC server waits for a TCP-timeout on a client socket
The VNC server displays GUI to all clients to notify them of the disconnection & disables all inputs on its sockets to halt the session
The VNC viewer reconnects after the disconnection
The VNC server enables the inputs of all clients and resume the session

Figure 7.5 Passive Client Approach

Another general method based on the same concept has been developed based on three main steps, which will be discussed here in accordance to the VNC system as an example of thin client architecture:

The VNC server performs a TCP-Timeout check every few seconds to know if a client has been disconnected.

Traditionally; TCP does not include a mechanism for probing idle connections. In theory, if a host crashes after establishing a connection to another host, the second machine will continue to hold the idle connection forever. Some TCP implementations include a mechanism that tests an idle connection and releases it if the remote host has crashed, TCP Keep-Alive is a mechanism periodically sends a probe segment to elicit response from the peer, if the peer responds to the probe by sending an ACK, the connection is alive. If the peer TCP fails to respond to probe segments from than a fixed threshold, the connection is declared done and connection is closed.

In general, this can be done by implementing a KEEP-ALIVE mechanism, According to RFC-1122 [218], a TCP implementation may include the keep-alive mechanism, however, if TCP keep-alive is included; the application must be able to turn it on and off. The threshold interval to send TCP keep-alive must be configured and must be configurable and must default to 7,200 seconds (two hours) or more.

If a server doesn't hear from a client, it could be because it has nothing to say, some network between the server and client may be down, the server or client's network interface may be disconnected, or the client may have crashed. Network failures are often temporary and TCP connections shouldn't be dropped as a result.

The purpose isn't to detect a crash immediately, but to keep unnecessary resources from being allocated forever.

Specifically for VNC, we can customise the vnc server to have a socket-timeout of 10 seconds (satisfaction threshold).

For example in vnc_4.1.1 in common/rfb/ServerCore.cxx

rfb::IntParameter rfb::Server:: idleTimeout //idle Timeout of 60 minutes

(IdleTimeout", "The number of seconds after which an idle VNC connection will be dropped " "(zero means no timeout)", 3600, 0);

rfb::IntParameter rfb::Server::clientWaitTimeMillis // default is 20 seconds

("ClientWaitTimeMillis","The number of milliseconds to wait for a client which is no longer " "responding",20000, 0); // for our scenario the 20000 should be replaced by 10000 ms (10 s).

And then invoke the Awareness message in figure 7.7

2. The VNC server displays a message to all VNC viewers to notify them of the coming disconnection of the client.

3. The VNC server then enforces the view only mode until the Mobile Node reconnects. Or VNC Server produces "synthetic" input events for the mouse & keyboard and so these can be completely filtered using an algorithm.

The advised change in VNC server code and VNC viewer code is to add an extra message from VNC client (a notification from the client to the server that a disconnection is going to occur). In addition to, Add an extra message for VNC server as a reply to the VNC viewer’s message (displaying a Screen with the words: one of the clients is disconnecting to satisfy the main goal of keeping the other members in the collaborative session informed with the status of each member’s presence and then the server informs them of its coming action of disabling their inputs (view only mode) so will not accept any inputs from the other clients until the disconnected client reconnects.

Also, one of the alternative solutions is going back to the proactive client approach and expecting the mobile node to perform the SNR mechanism to foresee a handover and before disconnecting sends a message to the VNC server to inform it of its possible disconnection.

7.4.5 IC-QoE Solutions: Notes on implementing IC-QoE Algorithm

7.4.5.1 The Awareness Mechanism

This prototype is based on the proactive client and is used as an explanatory evidence of the concept. To satisfy our scenario in 7.1 the steps are:

1. Collaborative session of three clients connecting to a shared session on a thin client server
2. A nomadic client using signal quality measures to predict a handover
3. before handover, the client press the send to server button to inform it of its impending disconnection, figure 7.6 as an example of GUI used on the nomadic client

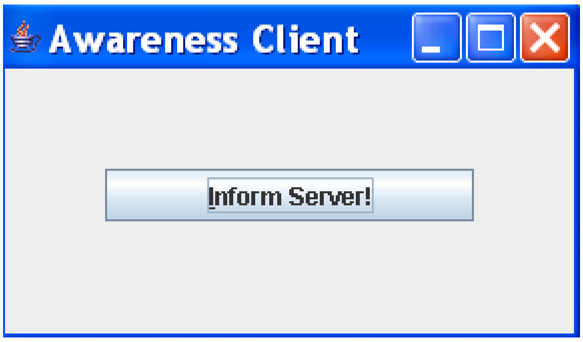
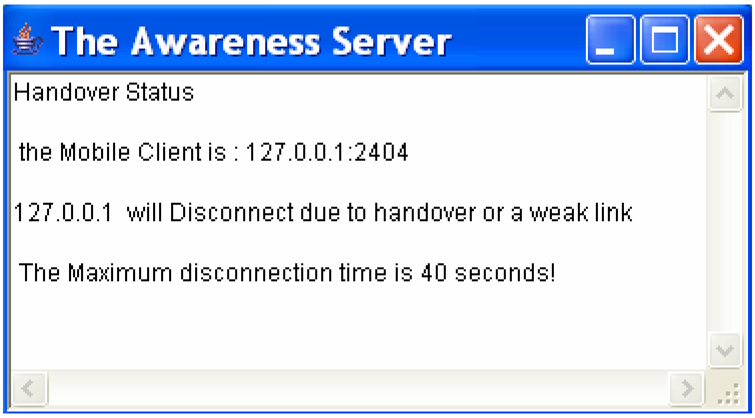


Figure 7.6 The Awareness client

4. The server will display the screen below shown in figure 7.7 with a message that will be relayed to all clients because of the vnc server's thin architecture. This message can be customised; here we display the disconnected client's IP address and the maximum number of seconds the clients should be waiting for him to return, for example 40 seconds as it is the frustration threshold.



```
Handover Status

 the Mobile Client is : 127.0.0.1:2404

127.0.0.1  will Disconnect due to handover or a weak link

 The Maximum disconnection time is 40 seconds!
```

Figure 7.7 The Awareness server

The source code of both the client and the server are in **Appendix C**

7.4.5.2 The Consistency Mechanism (Freeze-TCP)

First, the Freeze-TCP mechanism was incorporated by its developers for Linux kernel 2.1. After that an effort was made by [219] to adopt Freeze-TCP to kernel 2.4. The work done on freeze-TCP is based on the kernel 2.4 version. At the beginning I had access to a Suse 9.1 Linux based on kernel 2.6 and because of the tight schedule and short time needed to implement Freeze-TCP, attempts to migrate the source code of kernel 2.4 to the new kernel 2.6 was not successful due to the fact that Linux kernel 2.6 was a major upgrade from the earlier kernel 2.4 with many performance improvements and also Some techniques and APIs have been removed, and existing device drivers and modular plugins may no longer work as well as a change to many modules conventions which means the need to rewrite the freeze-tcp networking code to reflect those changes. The implementation of freeze-TCP and more of the implementation difficulties are in **Appendix D**.

I tested Freeze-TCP between two platforms, Redhat running the Freeze-TCP system call and a Linux SuSe 9.1, using an FTP server on Suse and transferring a large file between SuSe and Redhat and in the middle of the transfer I run the freeze.c program and the transfer was frozen for 10 seconds and then it resumed the transfer without disconnecting.

However, the compiled kernel with the Freeze-TCP capability, which went through modifications to the TCP/IP stack, was not stable to run a VNC Server on it even when tested on different hardware platforms such as a DELL desktop and a Toshiba Laptop. The kernel crashed each time a VNC viewer tried to connect to the VNC server whether on the desktop or the laptop. This did not happen on a non-modified kernel version, so we assumed that it is because of the instability of the kernel due to changed to the TCP/IP stack. This has limited the option of testing them together.

7.4.6 A Framework for measuring Improved Collaborative-QoE in the case of disconnections

QoE as mentioned before is a subjective metric that depends on the scenario of the task therefore it’s a challenge to specify how to measure it and to measure if it is improved.

For our scenario in figure 7.1 in the case of a disconnection of mobile node participating in a collaborative session, we could imagine that the quality of experience of disconnections is based on characteristics that affect collaborative environments both socially like awareness and consistency and technically like network metrics.

Hence I would like to suggest that an Improved Collaborative Quality of Experience is a function of social and technical characteristics, the overall term that encompass social interactions and depends on the most important characteristics of the collaboration such as awareness, consistency, transparency, security...etc is called the Quality of Collaboration (QoC). Where the overall term that defines the session characteristics that are based on the type of connection is called the Quality of Session (QoSn) and overall can be considered the sum of many factors such as:

1) Type of Session : Real-time, Non real-time
2) Frequency of disconnection
3) Type of disconnection: Weak link, handover
4) Link reliability: Packet loss before and after disconnection

Almost all characteristics are subjective and the only possibly measurable characteristic is link reliability.

Improved Collaborative-QoE = *f* (Quality of Collaboration, Quality of Session)

IC- QoE = *f* (QoC, QoSn)

Where

Quality of Collaboration = QoC = *f* (Collaboration characteristics)

Quality of Session = QoSn = *f* (Session Characteristics)

For our Scenario (Research team Meeting and Videoclip watching) :

1. **First we investigate Quality of Collaboration**
   - **QoC = *f* (Awareness, Consistency)**

QoC in the case of disconnection is a subjective term that prioritises two main characteristics in a collaborative session:

- Awareness of disconnection (other participants informed of the disconnection)

- Consistency of session before and after disconnection between all participants (introducing a delay, freezing the session or disabling inputs and outputs of all participants)

e.g. QoC = Awareness ∪ Consistency

Where

⇒ Consistency is a Boolean integer of 1 for freezing the session and 0 for a normal disconnected scenario

⇒ Awareness is a Boolean integer of 1 for informing all the participants of the disconnection of the mobile node

2. **Second we measure Quality of Session**
   - **QoSn= *f* (Link reliability)**

Quality of Session: is a function of link reliability

Link Reliability is a function of time because we measure the session loss in T

At T where T is the total disconnection time

And T= t2- t1 t1 is the time the mobile node disconnected and t2 is the time the mobile node re-connected

⇒ Average data transfer = ∑data transfer from server to client of all clients in T/total number of clients

⇒ Session loss = clients' session data transfer in T/n – data transfer of disconnected client in T

⇒ ∆ Session Loss = Session Loss without FTCP in T – Session Loss with FTCP in T

e.g. Link Reliability = Sign (∆ Session Loss) (2)

$$Link\,\mathrm{Re}\,liability = \begin{cases} -1, \Delta sessionLoss < 0 \\ 0, \Delta sessionLoss = 0 \\ 1, \Delta sessionLoss > 0 \end{cases}$$

If Δ Session Loss is positive then the QoSn = 1 increased

If Δ Session Loss it is zero there is no difference QoSn =0

If Δ Session Loss it is negative the QoSn= -1 decreased

3. **Finally, We substitute in the following equation:**
   - **IC-QoE = (QoC + QoSn) – 1**

$$QoE_{disconnection} = \begin{cases} -ve, IC - QoE < 0 \\ indifferent, IC - QoE = 0 \\ +ve, IC - QoE > 0 \end{cases}$$

If IC-QoE is 1 then QoE has improved

If IC-QoE is 0 then QoE is indifferent

If IC-QoE is -1 then QoE has decreased

8 Chapter Eight

Conclusions

8.1 Summary

As a conclusion, it is a fundamental for one seeking the analysis, design and development of an effective application to consider its target users. Moreover, it is reasonable to say that traditional Quality of Service (QoS) metrics such as response time and delay no longer suffice to fully describe quality of service as perceived by users. The success of any scheme that attempts to deliver desirable levels of QoS for the future Internet must be based, not only on the progress of technology, but on users' requirements. This introduces the term Quality of experience (QoE), which is a concept comprising all elements of a user's perception of the network and performance relative to expectation. For instance, a collaborative environment has not yet reached the desired expectation of its users since it is not capable of handling any unexpected exception which can result from a sudden disconnection of a nomadic user engaged in an ongoing collaborative session; this is consequently associated with breaking the social dynamics of the group collaborating in the session. Therefore I concluded that knowing the social dynamics of application's users as a group and their requirements and expectations of a successful experience can lead an application designer to exploit technology to autonomously support the initiating and maintaining of social interaction, rather than, supporting participants in their efforts to establish social interaction.

This thesis investigates VNC as a thin client and a collaborative environment and to develop a VNC based environment over the network to improve users' QoE between research groups or work teams collaborating distantly especially in the presence of sudden disconnections.

When adopting VNC for remote collaboration some interaction problems will rise such as the collision of using resources and not knowing whom are you sharing the desktop with. In the case of a team sharing the desktop for research purposes or to revise a paper or to test a prototype an additional channel can be introduced to overcome interaction problems such as chat software or telephone where every user can contribute to what is happening. There are problems in the feasibility of multi-user VNC environment such as the ID of the author of any modification cannot be recognised or known. Hence, there should be an awareness mechanism to inform users of the author of the actions. Also, clients lost synchronisation with the server due to the overload of events processed on the server from several modifications from multiple participants because of the no feedback mechanism of who done what and the confusion of the owner of the modification. Another weakness is the lack of access control to any resources to permit or deny the actions done upon them, as well as the lack of floor control where in a multi-user environment there are no policies of how to deal with the actions of several users at the same time and there are no methods of identifying the user who performed the actions.

Based on previous work done in Columbia University in comparing between different thin clients amongst them is VNC and looking at online documentation this thesis outlines the differences in performances between VNC, Citrix's Metaframe, Graphon's Go-Global, Windows Terminal Services and Tarantella Enterprise software.

VNC had been compared in [127] to two different thin client platforms and had been evaluated against them for its performance in supporting video applications: Citrix Metaframe and Microsoft Terminal Services. whereas in [126],[128] the WinVNC and Xvnc on Linux are used and compared to five thin client platforms which are Citrix Metaframe and Windows Terminal services again, Tarantella, SunRay[7] and X .

[7] Sun Ray 1 Enterprise Appliances. http://www.sun.com/products/sunray1

In [126] it is tested for both web and multimedia applications over a range of network bandwidth ranging from ISDN to a LAN environment while in [128] it evaluates the performance of the six thin client platforms in delivering computational services cross country over WAN.

In [127] Only the Windows version of VNC (WinVNC 3.3.3r2) was used for a fair testing against Metaframe and Terminal server, which both are Windows based solutions. The Benchmark used to measure web and multimedia performance is Windows Media Video benchmark from the Ziff-Davis i-Bench version 1.02 and was tested on Windows NT and 2K. At 100 mbps LAN network bandwidth the video clip playback was completed in 63 seconds on the three platforms however the video quality varied between them. As it is mentioned in [127] VNC delivered the worst video application performance where the results were equally bad on Windows NT and Windows 2K where the other two platforms delivered good video quality on both Windows NT and 2K. It was found that VNC's poor performance is most likely to be due to the poor implementation of Windows VNC version and not due to hardware resources limitation where [58] verifies that WinVNC is less robust than Xvnc. It was concluded from [127] that VNC delivers fairly constant poor video quality for different network bandwidths varying between 128 kbps rising up to 100mbps LAN environment. Moreover VNC offered the worst performance comparing to the other two platforms where little data was sent between the client and the server and although the video clip playback was completed in 63 seconds almost none of the frames were displayed on the client. An important point to consider is that existing thin-client solutions do not perform well for broadband network access bandwidths which is a disadvantage for current Application Service Providers (ASP).

A slow-motion benchmark was developed in [129] and was used in [126], [128] to provide an effective method of evaluating thin-client performance. In slow-motion benchmarking performance is measured by capturing network packet traces between a thin client and its respective server during the execution of a slow-motion version of a conventional benchmark application [129]. It was established that thin-client systems could provide good web performance in broadband or higher bandwidths yet not able to perform well on lower bandwidth environments such as ISDN and dial-up modems

[126]. Client-side network activity is monitored to obtain a measure of user-perceived performance based on network latency. Latency is the time it will take a single packet of data to travel to a remote server and return (round trip). A packet monitor is used to capture resulting network traffic on the client-side. Network latency can be measured by calculating the difference between when the user input is first sent from the client to the server and when the screen update finished sending from server to client. The time from the client input is made and the input is sent & the time from when the client receives a screen update from the network to the time the actual image is drawn to the screen are not included in the measurement. VNC was tested over Internet2 for web performance and video performance. The East and West sites were connected by Abilene Internet 2 backbone, an OC-48 operating at 2.5 Gbps[8]. The results in [128] show that VNC achieved the best web performance over Internet2 compared to the other thin-client systems yet it performs poorly on the video benchmark over both LAN and Internet2.

VNC uses demand-driven display update policy where the client requests the updates and a lazy display update model where multiple commands are merged before being sent to the client. This presents a problem for real-time video display because updates are not sent frequently enough for this purpose and since the client is loaded it becomes a bottleneck in requesting updates and this results in multiple video frames being merged and overwritten in the server before the client requests an update [126]. While lazy display updates is bandwidth efficient it is not suitable for multimedia applications such as video. The display update mechanism of thin-client systems is poorly suited to video applications.

VNC employs a 2D draw primitives and compression therefore the data transfer sent between the client and server is less than some other thin-client platforms. And VNC comes second after SunRay in Video quality. As a conclusion simpler pixel-based encoding methods offer better overall performance when sufficient network

[8] Abilene Weather Map http://hydra.uits.iu.edu/abilene/traffic/

bandwidth is available and network latency is what matters. VNC had the lowest latency over Internet2 than the other platforms.

Therefore, as a conclusion we can envision using multimedia in the future on VNC because multimedia systems are instrumental in the creation of a communication environment which allows remote working between design or development teams. Three technological fields can affect how people see using multimedia with VNC and help using VNC in the future: Advances in computing, Networking and compression techniques [63]. Where advances in computing provides more processing power per chip (CPU) and it is significantly in increase every year and the growth in storage capacity and progress in user interface and software concepts will help. The Advances in Networking affect the transmission of digitised multimedia information which already places a heavy requirement on the underlying networks. Also the progress in compression techniques affect using multimedia where compression refers to the algorithms used to reduce the bit rate of this signal by eliminating redundancies. Compression is necessary for two reasons as mentioned in [63]: Firstly, reduce the storage volumes of sound, images and motion video. Secondly, to limit the bit rate necessary to transmit them over networks. There are two kinds of compression: First, data compression (Lossless compression) which is the compression of computer-based data such as text, programs or scientific experimental result where the compression process can damage no data and decompression must recover all the data. Second, Lossy compression: it’s a mode where the uncompressed signal is different from the original signal.

VNC also is considered a remote control solution because it offers the user with the full control of a remote machine, therefore this thesis looked at four popular commercial remote control solutions such as NetSupport Manager, Altiris Carbon Copy, Unicenter Remote Control and pcAnywhere to derive the features that can be introduced to VNC to strengthen it as a remote control system and to provide maximum functionality to its customers. VNC already has powerful features such as the ability to share the desktop or workstation with other users hence this feature is very helpful in Computer Support and Virtual Classrooms. The feature that sets VNC in a strong position is the ability to view different kinds of hardware using a browser or a standard VNC viewer with the ability to disable the browser access, for example a Windows PC can view a Solaris workstation using a web browser or a VNC viewer downloaded from the Internet. In Xvnc, each user can have his/her own virtual session without interfering with any other session as well as sharing the same session with others when required.

Missing features in the original VNC are: a file transfer function to transfer files between the VNC server and viewer, this is implemented in a Windows-only VNC implementation called Ultra@VNC. Also VNC lacks the support for asynchronous collaboration such as recording the session for latecomers and people who join in after the session, a UNIX recording tools is been developed but not integrated in Xvnc. In addition to that it lacks printing redirection from the VNC server to print on the viewer printer yet printing to a file and sending it using FTP to the VNC viewer machine can overcome this. One desirable feature is to blank the server's screen when a viewer connects to it, so no one at the server's location can view the viewer's activities on the server to provide privacy for the user of the server. Ultra@VNC implements this blanking feature in addition to several other features such as a text chat facility. Also, VNC has a limited Copy and Paste support between the server and the viewer where it only copies ASCII text in each direction. To assist users to install VNC remotely, Fast push script and xVNC can be used to remotely deploy VNC for Windows platforms.

VNC introduces restlessness to network administrators due to the fact that it gives the connected viewer full access and control of the server. VNC uses a challenge/response method for authentication where the password is encrypted however after that the traffic is unencrypted which exposes it to network eavesdropping, this can be avoided using encryption packages such as SSH, SSL and Zebedee. Encrypting the traffic using SSH with VNC are implemented in this thesis.

One of the other problems is that if the user has a physical access to the VNC server he will be able to change the password manually yet to avoid rising suspicion of who maintains the server he can decrypt the existing password by retrieving it from the registry and decrypt it using simple dedicated tools tested in chapter five.

Also, the lack of any logging tools to record any actions performed represents a disadvantage for security conscious users though in both Xvnc and WinVNC there are files that records the IP addresses of who is trying to connect to them and who connected to them. It is fair to say that VNC also has powerful security features such as the ability to ask first the server's user if he permits the connection or rejects it, in addition to the capability of using IP authorisation to specify what IP addresses are permitted to connect to the server. A very interesting feature that is tested and implemented in this thesis is bypassing firewalls where the VNC server can initiate the connection from inside the firewall to any machine outside the firewall that is running a listening VNC viewer; another method is using tunnelling in SSH to overcome this problem. It is worth mentioning that the VNC version discussed here is 3.3.7 and in newer VNC 4.x versions especially the commercial personal and enterprise editions many of the security vulnerabilities were addressed and more secure features were introduced.

VNC is used in many environments for different purposes. It can be used by educational organisations to provide their students and staff access to any graphical applications on their UNIX workstations or Microsoft Windows. VNC also can be used as a part of a suite to facilitate distance learning to provide remote students with applications related to their area of discipline. for example, this eases any problems the students face in acquiring heavy software working on Linux while they use Windows and it can be used in a virtual classroom scenario where many students can view the desktop of their tutor so he can guide them through tutorials, this method will need another channel to provide audio or video communication for that purpose instead of only having a text guide, they need the tutor to explain in voice. VNC is also used in virtual laboratories to provide state of the art of equipments without the need to travel to the equipment itself.

Some work has been made in this thesis in the area of administration where four prototypes were developed to offer multi VNC sessions for users. The first two prototypes are based on HTA (HTML Applications) and Java Script/JScript owed to its widespread and portability where HTAs are applications trusted and contain the power of Internet explorer and Java Script is a script based programming language that supports the development of both client and server components of web-based applications. The difference between the two HTAs prototypes is the design of the GUI where one of them divides the screen vertically in half to provide two VNC sessions and it is not desirable to use more than two sessions otherwise they will be very narrow sessions and it would be impossible to carry out any work, in addition to that the session cannot be maximized therefore it will have vertical and horizontal scrollbars.

The second prototype divides the screen horizontally where the VNC session takes the whole width and height of the screen and to view the second VNC session the user scrolls down, this method overcomes the small display issues of the previous method and the two consecutive VNC sessions can be viewed and compared if needed. However, if more than two VNC sessions are included only every two sequential sessions can be viewed at once.

The Third prototype is based on an ActiveX control developed by Thong Nguyen called VNCX and this prototype divides the screen in half embedding VNCX and it overcomes the scaling problems in the Java version of VNC where zooming and scaling is provided by VNCX. Therefore the user is able to view two screens while scaling them too.

The fourth prototype is a Java application based on the Java VNC viewer where the GUI is the front end of having more than one viewer connecting to as many VNC servers where the administrator can switch from one to another.

In addition, I propose using videoconferencing in parallel to VNC to provide video and audio to the users of VNC and present videoconferencing technologies such as the H.323 standard which provides multimedia communication over networks that do not provide a quality of service and present SIP (Session Initiation Protocol) which is a signalling protocol for establishing calls and conferences over IP networks. And there are many tools developed to provide videoconferencing to the users of the Internet such as CUseeMe, vic, rat, Speak freely, NetMeeting, DC-share and SunForum Workgroup collaboration tools. In this thesis NetMeeting was chosen for its popularity on Windows and its functionality to be used in parallel with VNC and it was added to a VNC prototype.

According to [94, 104] the performance of real-time distributed groupware over real world wide-area networks has been frequently criticized. These performance problems are primarily due to network issues: latency, which is the time required for information to travel between locations; jitter, which is variance in latency; loss, which results from network packets not arriving at their destination; and insufficient bandwidth. These problems are common in today's wide-area networks, and although networking advances are aiming to reduce these problems, the problems will be present for some time to come. In the meantime, groupware applications must attempt to deal with these issues themselves.
Therefore, In the last chapter in the thesis, we considered the social and psychological effects of groups working together in a collaborative environments, the negative emotional states resulting from uncertainty and ambiguity in the collaborative environment can affect not only the interaction with the computer but also can cause frustration and communication breakdown between group members to the point that the frustration outruns the benefits of having the collaborative session and decreases the overall productivity. Therefore to reduce users' frustration, understanding users and collaborative application needs can allow us build an application that is aware of worst case scenarios and to enable them to function with acceptable quality over a network with limited performance characteristics like mobile networks.

Therefore, I benefited from research previously done in the HCI area and found that the common conclusion of many researches such as in [204, 1205, 206] is that the user of a computer dialog e.g. browsing a website , after a certain threshold (approximately 10 seconds), will be aware of time and therefore time becomes a factor in user satisfaction and slowly builds into annoyance and anything slower than 10 seconds needs a feedback to indicate if possible when the computer expects to be done [207] . Moreover, after the period of time between when a user is no longer satisfied and when he becomes significantly frustrated; the user enters the zone of frustration. Based on research in [205, 208, 209], we conclude that users experience significant frustration at approximately 30-40 seconds. Benefiting from these deductions to improve the QoE of users in a collaborative session specifically in the case when one client of them encounters disconnections due to his nomadic connection e.g. wireless or change of location, I propose using an improved collaborative QoE (IC-QoE), Two important characteristics of IC-QoE in disconnections is disconnection awareness with a feedback to all users, and disconnection consistency by providing a uniform session to all users by freezing the session.

IC-QoE can be either based on a passive client where it depends on the thin client server to discover the disconnection of one of its clients by using time-out mechanism after 10 seconds (satisfaction threshold). Or proactive where it depends on the client informing the server of disconnection after monitoring signals quality. The server displays a feedback to all users informing them of the disconnection of the client, the server freezes all of its TCP-connections until the client reconnects or for a maximum of 40 seconds (frustration threshold), and then the server unfreezes the session.

I implemented a prototype for awareness as well as the implementation of freeze-tcp approach to investigate freezing TCP connection. Freeze-tcp was chosen because it satisfies our requirement and its implementation is available and open source.

8.2 Future Work

Despite of VNC being a powerful tool, a lot of work can be done to enhance the Quality of Experience of VNC in a multi-user environment. Technology has made it feasible for many organisations to rely on geographically distributed teams. E-mail, fax and telephone do not adequately support such teams. Desktop conferencing, which combines information sharing with synchronous communication, holds promise for supporting virtual collocation. To use VNC in such environment will raise several challenges:

First, the goal of connecting remote team members is hard to achieve, due to the difficulty of adopting the technology especially if there was no one to consult and the ambiguity of the system will cause the resistance of using it.

Second, even after obtaining the technology running it smoothly will be required hence the need for an easy to follow documentation. When more than a user is using the technology a consultant is needed to answer questions and facilitate their usage.

Therefore in a multi-user environment it would be useful to investigate the challenge and problems presented above in the summary and one of these challenges is dealing with floor control. Integrating a floor control tool for a multi-user environment depends on what floor control policy they would like to adopt so for example in a chair control scenario where they elect a moderator who is responsible of handing over the session to who requests it. Where the moderator has a tool that lists all the clients connected referred to as numbers and IP addresses and he can hand in the control to the number corresponding to the client and cease its control too. As mentioned before a tool also must be developed to recognise all the pending floor requests. Another issue is who is attending the desktop and this can be known as mentioned before by listing the IP addresses of who is connected to the session using VNCWHO or netstat –a or using ultra@vnc on Windows which lists all clients in the VNC icon at the taskbar.

In relation to quality of experience in disconnections, many points can be further investigated such as the development of a collaborative environment prototype that integrates disconnection awareness and disconnection consistency mechanisms and test it in real-world scenarios.

Another area which could be investigated is the integration of sound with VNC where a way of pushing sound from the server to the viewer to hear what is playing from the soundcard at the server could be researched. Another way is developing a sound protocol parallel to VNC or implementing a sound thread attached to the VNC server. Also, more work and investigation can be done to optimise using video over VNC.

In the case of introducing videoconferencing with VNC, it is interesting to investigate using videoconferencing in a multipoint conference where Multipoint Conferencing Unit (MCU) is used to create virtual meeting rooms to connect three or more videoconferencing systems in the same conference and where several users are connected to the VNC server would like to be able to videoconference with the others. In H.323 standard it outlines two components responsible of any multipoint interaction: the MC (Multipoint Controller) and the MP (Multipoint Processor). The MC job involves forming connections between endpoints, negotiating common capabilities and communication to the MP regarding any necessary switching of audio/video resources. While MP handles the actual processing of incoming and outgoing audio/video streams. Where in a multipoint conference audio from all sites is typically mixed and delivered back to all sites in full duplex mode. Video on the other hand may be handled in a few different ways [193]:

1- Switched based on voice activation (everyone sees the current speaker)
2- Switched via manual control ("chair control", where the designated chair deciding whose video is being seen)
3- Displayed together on a split screen display ("continuous presence", also sometimes called "Hollywood squares")
4- Displayed individual video window, one for each site that is being received.

Hence it is interesting to try to implement any one of the above scenarios of video in a multipoint conference and a multi-user VNC environment. Another issue in a multipoint conference is multitasking to the desktop as a network service, which still seems to be facing a number of issues that is slowing down deployment especially over networks between organisations as apposed to within organisation. To host a multicast multipoint conference the host needs verification of the multicasting capability of each participating site. If one endpoint does not have multicast capability another means of enabling multipoint conference must be found to the multicast network such as tunnelling a unicast connection.

There are difficulties of deploying H.323 videoconferencing protocol [193] due to the complexity of H.323 protocol and its dependence on some of the utilised components (gatekeepers, MCUs and Gateways) and because of the widespread use of NAT and firewalls, therefore it would be useful to test passing the H.323 traffic through different types of firewalls such as the packet filtering, circuit-gateway and application proxy. The general solution is to open the required ports to all network traffic which will introduce significant security vulnerabilities into the internal system. Fortunately there are better alternative where many vendors have now implemented firewall technologies into router products that acknowledge the limitations of older implementations and properly support H.323 and SIP network traffic. These solutions can detect the videoconferencing signalling requests and take appropriate action to allow the traffic to traverse the router on the firewall.

Another area of future work is testing videoconferencing with VNC between different platforms and trying to connect between NetMeeting on Windows, Gnomemeeting on Linux and DC-share on Solaris and testing its feasibility and ease of use.

9 References

1. S. Lok, S. K. Feiner, W. M. Chiong and Y. J. Hirsch , “A Graphical User Interface Toolkit Approach to Thin-Client Computing”, In Proceedings International WWW Conference(11), Honolulu, Hawaii, USA. May 7-11, 2002. [Online]. Available: http://www2002.org/CDROM/refereed/577/
2. V. Kasacavage, "Complete book of remote access connectivity and security", Best Practises series, Auerbach publications, 2003.
3. A. Leon-Garcia, I. Widjaja, “Communication Networks: Fundamental concepts and key architectures”, second edition. McGraw-Hill
4. J. Postel. , “Transmission control protocol.”, IETF RFC 793, 1981.
5. R. Braden., “Requirements for internet hosts-- communication layers.”, IETF RFC 1122, October 1989.
6. M. Allman, V. Paxson, and W. Stevens., “TCP congestion control.”, IETF RFC 2581, April 1999.
7. J. F. Kurose and K. W. Ross , “Computer Networking: A Top-Down Approach Featuring the Internet”., 3rd ed. Addison Wesley.
8. P. Nixon and V. Cahill ,”Mobile Computing: technologies for a disconnected society”, Trinity College, Dublin, IEEE Internet Computing, January/February 1998
9. T. Richardson, F. Bennet, G. Mapp, and A. Hopper. “Teleporting in an X Window System environment”, IEEE Personal Communications, No.3, pp. 6-12, 1994.
10. M. Weiser, August 16, 1993 [online]. Available: http://www.ubiq.com/hypertext/weiser/UbiCompHotTopics.html,
11. A. dearle, "Towards Ubiquitous environment for mobile users", IEEE Internet Computing, Vol. 2, No. 1 pp. 22-32, January/February 1998.

12. F. Bennett, T. Richardson, A. Harter," Teleporting - Making Applications Mobile", Proceedings of 1994 Workshop on Mobile Computing Systems and Applications, Santa Cruz, December 1994

13. F. Stajano, "Security for ubiquitous computing", John Wiley & sons; 1st edition June 15, 2002

14. C. Perkins, " IP Mobility Support"., IETF RFC2002 ,October 1996

15. P. Marichamy, S. Chakrabati and S. L. Maskara, "Overview of handoff schemes in cellular mobile networks and their comparative performance evaluation", IEEE Vehicular Technology Conference , vol. 3, pp. 1486-1490, 1999.

16. A. S. Tanenbaum, "Computer Networks", Fourth Edition, Prentice Hall, 2003.

17. A. Noerpel and Yi-Bing Lin, "Handover Arrangement for a PCS Network", IEEE Personal Communications, vol. 4, pp. 18-24, December 1997

18. N. D. Tripathi, J. H. Reed and H. F. VanLandinoham, "Handoff in Cellular Systems", IEEE Personal Communications, vol. 5, pp. 26-37, December 1998.

19. C. Blondia, O. Casals, L. Cerdà, N. Van den Wijngaert, G. Willems, "Performance Evaluation of Layer 3 Low Latency Handoff Mechanisms", Mobile Networks and Applications, Vol. 9, No. 6 , pp. 633-645, December 2004

20. L. Yu, Y. Min-hua; Z Hui-min; "The handoff schemes in mobile IP", Vehicular Technology Conference,The 57th IEEE Semiannual Vol.1, 22-25 pp.485 - 489 , April 2003

21. H. Balakrishnan, "challenges to reliable Data transport over heterogeneous wireless networks", PhD thesis, University of California at Berkeley, Aug 1998

22. V. Jacobson. "Congestion avoidance and control." in Proceedings of the Symposium on Communications Architectures and Protocols (SIGCOMM), Stanford, CA, pp. 314--329, August 1988.

23. J. Schiller, "Mobile Communications", Addison Wesley; 2 edition ,August 29, 2003

24. V. Paxson and M. Allman. "Computing TCP's retransmission timer.", IETF RFC 2988, November 2000.

25. W. Stevens. "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms.", IETF RFC 2001, Jan. 1997.

26. P. Jain, R. Kelkar, “ Mobile IP: Enabling Mobility for the 3G Wireless Internet”, 2003

27. T. Alanko, M. Kojo, H. Laamanen, M. Liljeberg, M. Moilanen, and K. Raatikainen. “Measured performance of data transmission over cellular telephone networks.” Computer Communications Review, Vol. 24 No. 5. pp. 24–44, October 1994.

28. P. Karn. “The Qualcomm CDMA digital cellular system”. In Proceedings of the USENIX Mobile and Location-Independent Computing Symposium, p.p 35–39, August 1993.

29. S. Nanda, R. Ejzak, and B.T. Doshi. “A retransmission scheme for circuit-mode data on wireless links.” IEEE Journal on Selected Areas in Communications, Vol. 12 No.8. pp. 1338–1352, October 1994.

30. G. Xylomenos, G. C. Polyzos, P. Mähönen, M. Saaranen, “TCP/IP Performance over Wireless Networks”, Published in: High Performance TCP/IP Networking, Prentice Hall, 2002

31. A. Gurtov , “Effect of Delays on TCP Performance” , Proceedings of IFIP Personal Wireless Communications '2001, August 2001, Lappeenranta, Finland

32. R. Ludwig., “Eliminating Inefficient Cross-Layer Interactions in Wireless Networking.” PhD thesis, Aachen University of Technology, April 2000.

33. R. Ludwig, B. Rathonyi, A. Konrad, K. Oden, and A. Joseph. “Multi-layer tracing of TCP over a reliable wireless link.” In Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computing Systems (SIGMETRICS-99), volume 27, 1 of SIGMETRICS Performance Evaluation Review, pp. 144--154, New York, May 1—4 1999. ACM Press.

34. G. Brasche and B. Walke. “Concepts, services and protocols of the new GSM phase 2+ general packet radio service.”, IEEE Communications Magazine, pp. 94--104, August 1997.

35. R. Ludwig and R. H. Katz. “The Eifel algorithm: Making TCP robust against spurious retransmissions.”, ACM Computer Communication Review, Vol. 30 No.1, January 2000. Available at:

http://www.acm.org/sigcomm/ccr/archive/2000/jan00/ccr-200001-ludwig.html.

36. V. Jacobson, C. Leres, and S. McCanne. tcpdump. Available at http://ee.lbl.gov/, June 1997.

37. E. Hossain, N. Parves, "Wireless communications systems and networks", New York, USA: Plenum Press,2004, pp. 241-289

38. H. balaKrishnan, V. padmanbhan, S. Seshan and R. H. Katz " a comparison of mechanisms for improving TCP performance over wireless links", ACM SIGCOM'96, Aug 1996

39. H. balakrishnan, S. Seshan , R.H Katz,"improving reliable transport and handoff performance in cellular wireless networks," Wireless Networks, J.C Baltzer, no. 1., 1995

40. E. A. Brewer, R. H. katz, Y. Chawathe, S. D. Gribble, T. Hodes, G. Nguyen, M. stemm, T. Henderson, E. Amit, H. balakrishnan, A. Fox, V. Padmanabhan, S. Seshan, "a network architecture for heterogeneous mobile computing," IEEE Personal Communications, Vol. 5 No.5., 1998

41. A. Bakre, B. Badrinath, "I-TCP:Indirect-TCP for mobile hosts", proc. Fifteenth international conference on distributed computing systems (ICDCS'95), pp 136-143, 1995

42. K. Brown, S. Singh, "M-TCP:TCP for mobile cellular networks" ACM computer communications review, Vol.27 No.5 ,1997.

43. T. Geof, G. Moronski, and S. Phatak. "Freeze-TCP: A True end-to-end Enhancement Mechanism for Mobile Environments". In Proceedings of the INFOCOM, 2000.

44. F. Chai, "Micro-Mobility Management and Freeze-TCP Implementation in a Wireless Network", ME THESIS ,Electrical & Computer Engineering Dept. University of Canterbury, New Zealand, 2004

45. D. E. Comer, "Computer Networks & the Internets with Internet applications", Third Edition, New Jersey, USA :Prentice Hall, 2001, p.399

46. Mark Smith, "Thin Is In", Windows & NT Magazine, September, 1997, [Online].Available: http://www.win2000mag.com/Articles/Index.cfm?ArticleID=521

47. Newburn, whitepaper on Thin client benefits, version 1a, 27 March 2002, [Online] Available: http://www.wyse.com/resources/whitepapers/

48. S. Greenberg, C. Anderson, J. M. Jackson, "power to the people: comparing power usage for PCs and thin clients in an office network", [Online]. Available: http://www.thinclient.net

49. The X Windows System [Online]. Available: http://www.x.org

50. R. W. Scheifler and J. Gettys, "X window system: the complete reference to Xlib, X Protocol, ICCCM, XLFD". Bedford, Mass: Digital Press, 3rd ed, 1999.

51. Window Managers for X [Online]. Available: http://www.plig.org/xwinman/

52. T. Richardson, "Teleporting—Mobile X Sessions," Proc. 9th Ann. X TechnicalConf., Jan. 1995. Also in The X Resource, Issue 13, O'Reilly & Associates, Jan. 1995.

53. H. M. Abdel-Wahab and M. A. feit, "XTV: a framework for sharing X windows clients in Remote synchronous collaboration", Proceedings of IEEE TriComm 91: Communications for distributed Applications and systems, chapel Hill, North Carolina, April 1991

54. An archive of AT&T laboratories Cambridge [Online]. Available: http://www.cl.cam.ac.uk/Research/DTG/attarchive/

55. K. R. Wood, T. Richardson, F. Bennett, A. Harter and A. Hopper, "Global Teleporting with Java: Toward Ubiquitous Personalized Computing," Computer, Vol. 30, No. 2., pp. 53-59, Feb. 1997

56. T. Richardson, Q. Stafford-Fraser, K. R. Wood and A. Hopper, "Virtual Network Computing", IEEE Internet Computing, Vol. 2 No. 1, January/February 1998

57. S. F. Li, Q. Stafford-Fraser and A. Hopper, "Integrating Synchronous and Asynchronous Collaboration with Virtual Network Computing", 2000

58. RealVNC [Online] Available: http://www.realvnc.com/documentation.html

59. R. S Fish, R. E. Kraut and M. D. P. Leland, "Quilt: A collaborative tool for cooperative writing." In Conference on office information systems, Palo Alto, CA, USA, pp. 30-37, March, 1988.

60. C.A. Ellis, S.J. Gibbs and G.L. Rein, “Design and use of a group editor”. In Engineering for human-computer interaction, North-Holland, Amsterdam, NL, p. 13-28, 1990.

61. M. Koch, “Multiuser editor list (long)”, July 18, 1994 [originally posted to UseNet newsgroup comp.groupware], [Online] Available: http://www11.informatik.tumuenchen.de/cscw/multiusereditor.htm

62. M. Stefik, D. G. Bobrow, G. Foster, S. Lanning, and D. Tatar., “WYSIWIS revisited: Early experiences with multiuser interfaces”. ACM transactions on office information systems, Vol. 5, No. 2, p. 147-167, April 1987.

63. F. Fluckiger, "Understanding networked multimedia: Applications and technology", London: Pearson education. P142-144

64. J. Grudin, S. E. Poltrock, "CSCW & Groupware”, Advances in Computers Vol. 45 (pp. 269-320). Orlando, FL: Academic Press, 1997

65. R. M. Baecker, D. Nastor, I.R. Posner and and K.L. Mawby., “The user-centred iterative design of collaborative writing software”, Human factors in computing systems: INTERCHI'93. ACM Press, New York, NY, USA, 1993, p. 399-405.

66. B. Corrie, H. Wong, T. Zimmerman, S. Marsh, A.S. Patrick, J. Singer, B. Emond, & S. Noël. “Towards quality of experience in advanced collaborative environments.” Paper presented at the Third Annual Workshop on Advanced Collaborative Environments, Seattle, Washington, June 22, 2003

67. P. Dourish, "The Parting of the Ways: Divergence, Data Management and Collaborative Work," in Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work (ECSCW'95), Stockholm, Sweden, 1995.

68. G. H. Ter Hofte, “Generic service features of CSCW applications: An analysis of coauthoring tools “ (Platinum Deliverable D2.2/011: PLATINUM/N008/V00). Telematics Research Centre, Enschede, the Netherlands, 1996.

69. D. Miras, “A Survey of Network QoS Needs of Advanced Internet Applications”, Working Document, Computer Science Department, University College London, 2002.

70. S. Greenberg, and D. Marwood, “Real time groupware as a distributed system: Concurrency control and its effect on the interface”. In CSCW'94 : Proceedings of the conference on computer supported cooperative work, Chapell Hill, NC, USA, October 22-26, 1994.

71. C. Gutwin and S. Greenberg. “Workspace Awareness for Groupware.” Proceedings of the CHI'96 Conference Companion on Human Factors in Computing Systems, pp. 208-209, 1996.

72. S. Brennan, “Seeking and Providing Evidence for Mutual Understanding,” Ph.D. thesis, Stanford University, Stanford, CA, 1990.

73. H. Clark, “Using Language”, Cambridge University Press, Cambridge, 1996

74. E.J. Patrick, "Barriers to collaboration: User-centered research and the Access Grid." AccessGrid Technical Retreat Proceedings, Argonne National Laboratory, January 30-31 2001.

75. C. V. Bullen, J. L. Bennett, “Groupware in Practice: An Interpretation of Work Experience.” In Proceedings of the Conference on Computer Supported Cooperative Work, Los Angeles, CA, ACM/SIGCHI, NY, pp. 291-302, October 1990.

76. W.J. Orlikowski, “Learning from Notes: Organizational Issues in Groupware Implementation.” Proceedings of the Conference on Computer Supported Cooperative Work, Toronto, Canada , ACM/SIGCHI & SIGOIS, NY, , pp. 362-369, November 1992

77. M. L. Markus, “Towards a "Critical Mass" Theory of Interactive Media: Universal Access, Interdependence and Diffusion. “, Communication Research, , Vol. 14, No. 5, pp. 491-511, 1987

78. M.L. Markus, and T. Connolly, “Why CSCW Applications Fail: Problems in the Adoption of Interdependent Work Tools.” In Proceedings of the Conference on Computer Supported Cooperative Work, Los Angeles, CA, ACM/SIGCHI & SIGOIS, NY, pp. 371-380. October 1990.

79. J.Grudin, “Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces.” In Proceedings of the Conference on

Computer Supported Cooperative Work, Portland, OR, ACM/SIGCHI & SIGOIS, NY, pp. 85-93. September 1988.

80. J. Grudin, "Groupware and social dynamics: Eight challenges for developers." Communication of the ACM. Vol 37, No 1, pp. 92-105, January 1994

81. J. Grudin, "Obstacles to user involvement in software product development, with implications for CSCW." International Journal of Man-Machine Studies, Vol. 34, No. 3, pp. 435-452, 1991

82. B. M. Johnson, and R. E. Rice, "Managing Organizational Innovation: The Evolution from Word Processing to Office Information Systems.", New York, NY: Columbia University Press, 1987.

83. A. P. Strassman, "Information Payoff: The Transformation of Work in the Electronic Age." The Free Press, New York, NY, 1985

84. A. Acharya, and J. Saltz, "A Study of Internet Round-Trip Delay." Report CSTR-37-36, Dept. of Computer Science, University of Maryland, 1996

85. C. Gutwin, "the Effects of Network Delay on Group Work in Shared Workspaces," Proceedings of the European Conference on Computer-Supported Cooperative Work, Germany, pp, 299-318, 2001

86. K. Park, and R. Kenyon, "Effects of Network Characteristics on Human Performance in the Collaborative Virtual Environment". Proceedings of IEEE Virtual Reality '99, Texas, USA, March 14-17, p.104 , 1999

87. C. Gutwin, J. Dyck, J. Burkitt, " Using Cursor prediction to smooth telepointer Jitter", proc. ACM Group, Florida, USA, pp. 294-301, 2003.

88. F. Scholl, "Planning for Multimedia." White paper, Monarch Information Networks, 2000. [Online]. Available: http://www.monarch-info.com/pmm.html

89. Cisco Corporation, "Designing Internetworks for Multimedia.", 2000, [Online]. available:
www.cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2013.htm

90. C. Gutwin, R. Penner, "Improving interpretation of remote gestures with telepointer traces", Proceedings of the ACM conference on Computer supported cooperative work , Louisiana, USA, pp. 49-57, 2002.

91. J. Dyck, C. Gutwin, S. Subramanian, and C. Fedak, “High-Performance Telepointers,” Proceedings of the ACM Conference on Computer-Supported Cooperative Work, Illinois, USA ,2004.

92. C. Gutwin, and S. Greenberg, “Effects of Awareness Support on Groupware Usability”. Proceedings of ACM CHI'98, Los Angeles, USA, ACM Press, 1998.

93. S. Greenberg, and D. Marwood, “Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface”, Proceedings of the Conference on Computer-Supported Cooperative Work, pp. 207-217, 1994.

94. I. Vaghi, C. Greenhalgh, and S. Benford, “Coping with Inconsistency due to Network Delays in Collaborative Virtual Environments”. Proceedings of the ACM Workshop on Virtual Reality and Software Technology,London, UK, pp. 42-49, 1999.

95. C. Greenhalgh, S. Benford, G. Reynard, “A QoS architecture for collaborative virtual environments”, Proceedings of the seventh ACM international conference on Multimedia (Part 1), Orlando, Florida, United States , p.121-130, October 30-November 05, 1999,

96. A. Quintero, S. Pierre, and D. Tassy, “A Collaborative Learning Environment Architecture Supporting Quality of Service.”, Advanced Technology for Learning Vol. 3, No. 1, 2006

97. Y. Yan, Y. Liang, X. Du, H. Saliah-Hassane, A. Ghorbani, "Design Instrumental Web Services for Online Experiment Systems", World Conference on Education Hypermedia and Telecommunication, Montreal,Canada, June 27-July 2, 2005 .

98. M. Alfano, R. Sigle, "Controlling QoS in a Collaborative Multimedia Environment," hpdc, p. 340, Fifth IEEE International Symposium on High Performance Distributed Computing (HPDC-5 '96), 1996.

99. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. “An architecture for differentiated services,” IETF RFC 2475, Dec. 1998.

100. S. Shenker, C. Partridge and R. Guerin. “Specification of Guaranteed Quality of Service.”, IETF RFC 2212, Sep. 1997.

101. S. Cheshire, “Latency and the quest of interactivity,” [Online], Available : http://www.stuartcheshire.org

102. V. Strumpen, "Software-Based Communication Latency Hiding for Commodity Workstation Networks," International Conference on Parallel Processing (ICPP'96) , Volume 1, Ithaca, NY, USA p. 0146, 1996.

103. T. M. Warschko, C. G. Herter, and W. F. Tichy, "Latency hiding in parallel systems: A quantitative approach." Internal Report 10/94, University of Karlsruhe, School of Computer Science, March 1994. 8

104. C. Gutwin, S. Benford, J. Dyck, M. Fraser, I. Vaghi, and C. Greenhalgh. , "Revealing Delay in Collaborative Environments." In: ACM Conference on Computer-Human Interaction (CHI'04), Vienna, Austria, pp. 503--510. ACM Press, April 2004.

105. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture." RFC 3031, Jan. 2001.

106. L. Faucheur, R. Uppili, A. Vedrenne, P. Merckx and T. Telkamp ,"Use of Interior Gateway Protocol (IGP) Metric as a second MPLS Traffic Engineering (TE) Metric", IETF RFC 3785, May 2004

107. D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao, "Overview and Principles of Internet Traffic Engineering", IETF RFC 3272, May 2002.

108. M. Stiller and J. Woods, "Improving Quality of Experience for Multimedia Services by QoS Arbitration on a QoE Framework", IEEE PV 2003 Proceedings, 13th packet video workshop, Nantes, France, April 2003

109. B. Bauer, and A. S Patrick, "A human factors extension to the seven-layer OSI reference model." Retrieved January 6, 2004, unpublished. [Online]. Available: http://www.andrewpatrick.ca/OSI/10layer.html

110. L. Alben, "Quality of experience: Defining the criteria for effective interaction design." Interactions, Vol. 3 no.3, 1996, pp. 11-15

111. VNCWHO[Online].Available: http://www.sysworksoft.net/products/vncwho.html

112. S. F. Li, Q. Stafford-Fraser and A. Hopper, "Applications of stateless client systems in collaborative enterprises", Proceedings of the 1st International Conference on Enterprise Information Systems, Setubal, Portugal, pp. 665-673,1999

113. S. F. Li, and A. Hopper, “What You See Is What I Saw: Applications of Stateless Client Systems in Asynchronous CSCW”, The Fourth International Conference on Computer Science and Informatics (CS&I’98), Research Triangle Park, North Carolina, Oct.23-28, 1998.

114. S. F. Li, M. Spiteri, J. Bates and A. Hopper, “Capturing and Indexing Computer-based Activities”, Proceedings, 2000 ACM Symposium on Applied Computing , Como, Italy, 19-21 March, 2000.

115. S. F. Li, Q. Stafford-Fraser and A. Hopper, "Frame-buffer on Demand: Applications of stateless Client Systems in Web-based Learning", Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis (ISAS'99), Orlando, Florida, July 1999.

116. S. F. Li, Q. Stafford-Fraser and A. Hopper, "Integrating Synchronous and Asynchronous Collaboration with Virtual Network Computing", Proceedings of the First International Workshop on Intelligent Multimedia Computing and Networking, Atlantic City, USA, Volume 2, Pages 717-721, February 2000

117. I. Yihji, J. H. S. Yang, "Toward Better Assessment in Distance Education", 21st International Conference on Distributed Computing Systems Workshops, Mesa, Arizon, 16-19 April 2001

118. Windows VNC authentication [Online]. Available:
http://www.floydsoft.com/vncauth.html

119. VNC FAQ-o-Matic, “Setting up VNC on Linux” [Online]. Available: http://faq.gotomyvnc.com

120. J. P. Kanter, “Understanding Thin-Client/Server Computing", Microsoft Press, 1998

121. Citrix [Online]. Available: http://www.citrix.com

122. GraphOn Go-Global Software [Online]. Available:
http://www.graphon.com/

123. Microsoft Terminal Server, “Technical overview of Terminal Services.” July 2002, [Online] Available:
http://www.microsoft.com/windowsserver2003/techinfo/overview/termserv.mspx

124. Remote Desktop Protocol (RDP) Features and Performance, white paper, [Online]. Available: http://www.microsoft.com/windows2000/techinfo/howitworks/terminal/rdpfandp.asp

125. A tarantella white paper, July 2002 [Online]. Available: http://www.tarantella.com/whitepapers

126. S. J. Yang, J. Nieh, M. Selsky, and N. Tiwari, "The Performance of Remote Display Mechanisms for Thin-Client Computing". In Proceedings of the 2002 USENIX Annual Technical Conference, Monterey, CA, USA, June 2002.

127. J. Nieh and S. J. Yang. "Measuring the Multimedia Performance of Server-Based Computing". In Proceedings of the 10th International Workshop on Network and Operating System Support for Digital Audio and Video, Chapel Hill, NC, June 2000.

128. Lai and J. Nieh, "Limits of Wide-Area Thin-Client Computing". In Proceedings of the ACM SIGMETRICS 2002, Marina del Ray, CA, June 15-19, 2002

129. S. J. Yang, J. Nieh, and N. Novik. “Measuring Thin-Client Performance Using Slow-Motion Benchmarking". In ACM Transaction on Computer systems, pages 87-115, Vol. 21, No. 1, February 2003.

130. NetSupport Manager [Online]. Available: http://www.netsupport-inc.com

131. Altiris Carbon Copy [Online]. Available: http://www.altiris.com/products/carboncopySolution.aspx

132. Unicenter Remote Control [Online]. Available: http://www3.ca.com/Solutions/Product.asp?ID=228

133. pcAnywhere [Online]. Available: http://www.symantec.com/

134. David storm, "Control everything" , Network World Fusion, 20 August 2001. [Online]. Available: http://www.nwfusion.com/reviews/2001/0820rev.html

135. Ultra@VNC [Online]. Available:http://ultravnc.sourceforge.net/

136. FastPush[Online]. Available: http://www.darkage.co.uk/

137. Zvnc [Online]. Available: http://home.comcast.net/~davedyer/znc/zvnc.html

138. C. Brenton and C. Hunt, "Mastering™ Network Security", Second Edition. CA, USA : SYBEX Inc ,October 2002

139. C. Lynch, "A whitepaper on authentication and access management issues in cross-organizational use of networked information resources, Coalition for Networked Information" , April 1998. [Online]. Available: http://www.cni.org/projects/authentication/authentication-wp.html,

140. B. Pfaffenber, "Webster's new world Computer Dictionary", 10th Edition, 2003

141. S. McClure, J. Scambray, G. Kurtz, "Hacking Exposed: network secrets and solutions ", third edition , USA: Osborne/McGraw-Hill, October 2001

142. SnadBoy Software [Online]. Available: http://www.snadboy.com

143. Securiteam beyond security [Online]. Available: http://www.securiteam.com/

144. VncDecryption[Online].Available: http://packetstormsecurity.org/Crackers/vncdec.c

145. Div-C++ [Online]. Available: http://www.bloodshed.net

146. E. Kargieman, A. Azubel and M. Caceres from Core SDI, "Vulnerability Report For Weak authentication in ATT VNC", 15 Nov 2001. [Online].Available: http://www1.corest.com/common/showdoc.php?idxseccion=10&idx=117

147. PuTTy: SSH Client [Online]. Available: http://www.chiark.greenend.org.uk/~sgtatham/putty/

148. "Security COMPLETE", Second Edition, 2002, Sybex Inc, ISBN: 0-7821-4144-7

149. Stunnel Binaries [Online]. Available: http://www.stunnel.org/download/binaries.html

150. Artur Maj, "Remote desktop management solution for Microsoft". [Online]. Available: http://www.securityfocus.com/infocus/1677

151. ZeBeDee Encryption [Online]. Available: http://www.winton.org.uk/zebedee/download.html

152. M. Strebe and C. Perkins, "Firewalls 24seven™", Second edition, Sybex Inc, 2002.

153. Firewall VNC client [Online]. Available:

http://www.xs4all.nl/~harmwal/vnc/readme.html

154. rvnc: simple vnc-proxy across masquerading firewalls [Online]. Available: http://www.realvnc.com/pipermail/vnc-list/1998-June/001768.html

155. University of Auckland [Online]. Available: http://web.com.auckland.ac.nz/services/

156. University of Newcastle upon Tyne [Online]. Available: http://www.ncl.ac.uk/

157. University of Aberdeen [Online]. Available: http://www.abdn.ac.uk/diss/compserv/class/vnc

158. University of Manitoba [Online]. Available: http://www.umanitoba.ca/campus/acn/support/vnc/web2unix.html

159. Heriot-watt University [Online]. Available: http://www.sml.hw.ac.uk/computing/runvnc.html

160. University of Nebraska-Lincoln[Online].Available: http://aac.unl.edu/virtual.html

161. D. Agarwal, "Collaborating across the Miles", the Proceedings of the INMM/ESARDA workshop on science and modern technology for safeguards, Albuquerque, NN, September 22, 1998

162. W.Wulf, "The national collaboratory-A white paper", In J.Lederberg, and K. Uncapher (Eds) Towards a National Collaboratory: Report of an Invitational Workshop at the Rockefeller University, march 17-18 1989

163. J.P Vary (Ed.), "Report of the Expert Meeting on Virtual Laboratories" ,International Institute of Theoretical and Applied Physics (IITAP) Ames, Iowa,10-12 May, 1999

164. University of Chalmers LindHolmen [Online] : http://www.chl.chalmers.se/~numa/

165. M. Johnston, F. Cox, G. Forte, D. Nairn, R. Sacher, A. Schwartz and A. Vertes. ,"Remote Experimentation over the Net: our first year with MALDI", Analytical chemistry, Vol. 73, No.15, pp. 440-445 A. ACS publications, August 2001.

166. S. McCracken, Z. Zilic and H. Chan, "Real Laboratories for Distance Education", Journal of Computing and Information Technology, Vol. 11, No. 1, pp67-76, June 2003.

167. L. W. Pettersson, N. Jensen, S. Seipel, "A Virtual laboratory for Computer Graphics Education" , (Sweden), Eurographics 2003/ S.Cunningham and D.Martin,[Online]Available:http://projekte.learninglab.uni-hannover.de/pub/bscw.cgi/d29096/pettersson_virtual_laboratory.pdf

168. OpenGL [Online] Available: http://www.opengl.org/

169. S. Stegmaier, M. Magallon and T. Ertl, "A Generic Solution for Hardware Accelerated Remote Visualization", Joint Eurographics, IEEE TCVG Symposium on Visualization 2002, D. Ebert, P. Brunet, I. Navazo (Editors).

170. A. Berqia, A. Diop, J. Harms ,"A Virtual Laboratory for Practical Exercises" , International Conference on Engineering Education, Manchester, U.K., August 18-21, 2002.

171. WebCT [Online] Available: http://www.webct.com

172. Rembo Technology [Online] Available: http://www.rembo.com

173. Webmin [Online] Available : http://www.webmin.com

174. R. J. Figueiredo, J. A. B. Fortes, R. Eigenmann, N. H. Kapadia, S. Adabala, J. Migul-Alonso, V. Taylor, M. Livny, L. Vidal, and J. Chen, "A Network-Computing Infrastructure for tool experimentation applied to computer architecture education",Workshop on Computer Architecture Education at the 27th Annual International Symposium on computer architecture (ISCA'2000), June 2000, Vancouver, Canada.

175. Cadence Design Environment in New Mexico State University. [Online]. Available: http://www.ece.nmsu.edu/vlsi/cadence.html

176. S. Basu, V. Talwar, B. Agarwalla, R. Kumar, "Interactive Grid Architecture for Application Service Providers", International conference on web services (ICWS '03), 23-26 June 2003, Las Vegas, Nevada.

177. A. Hasedawa, T. Nakajima, Waseda University, "A User Interface system for home appliances with Virtual Network Computing", 21st International

Conference on Distributed Computing Systems Workshops, Mesa, Arizona, 16-19 April 2001

178. uVNC. [Online]. Available: http://www.sics.se/~adam/uvnc/

179. PalmVNC [Online].Available: http://www.btinternet.com/~harakan/PalmVNC

180. PalmVNC 2.0 [Online]. Available: http://palmvnc2.free.fr/

181. VNC viewer for PocketPC [Online]. Available: http://www.cs.utah.edu/~midgley/wince/vnc.html

182. J2ME VNC [Online]. Available: http://j2mevnc.sourceforge.net/

183. VNC server for Windows CE [Online]. Available: http://sourceforge.net/projects/wincevncsvr/

184. Zaurus framebuffer VNC server [Online]. Available http://zsi2.stonekeep.com/index.php?v=d&a=21

185. VNCCAM [Online]. Available: http://vnccam.sourceforge.net/

186. VNC viewer for Nokia Communicator [Online]. Available: http://www.pdacity.com/apps/details.asp?AppID=63

187. B. Shizuki, M. Nakasu and J. Tanaka, "VNC-based access to remote computers from cellular phones", Proceedings of the IASTED International Conference on communication systems and networks (CSN 2002), Benalmádena, Malaga, Spain, pp. 74-79, September 9-12,2002.

188. Apache Tomcat 4.0 [Online]. Available: http://www.apache.org

189. J. Jaworski, "Mastering JavaScript and Jscript: The essential Reference for Building Interactive Web Pages", Sybex inc., 1999

190. P.Campbell, B. Calvert, S. Boswell, "Security+ Guide to Network Security Fundamentals", Cisco Learning Institute 2003.

191. VNCX- VNC ActiveX Control [Online] Available: http://www.veridicus.com/tummy/programming/vncx/default.asp

192. "The Java™ Platform", A White Paper By Douglas Kramer With contributions by Bill Joy and David Spenhoff, May 1996. [Online] Available: http://java.sun.com/docs/white/platform/javaplatformTOC.doc.html

193. Video Development Initiative, “ViDe Video Conferencing Cook Book”, February 2004 [Online] Available: http://www.videnet.gatech.edu/cookbook/

194. T.120 standard [Online] Available: http://www.microsoft.com/windows/NetMeeting/Corp/reskit/Chapter10/default.asp

195. CUseeMe [Online] Available: http://www.cuworld.com/

196. Video Conferencing Tool [Online] Available: http://www-nrg.ee.lbl.gov/vic

197. Robust Audio Tool [Online] Availabe: http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/

198. Floyd Jay Winters and Julie T. Manchester, “Web Collaboration Using office XP and NetMeeting”, prentice hall 2002

199. SunForum Workgroup Collaboration Tools: [Online] Available: http://www.sun.com/desktop/products/software/sunforum/

200. J. Balrow, P. Peter and L. Barlow, “Smart Videoconferencing: New Habits for Virtual Meetings”, First Edition, san Francisco: Berret-koehler publishers, 2002

201. L. M. Schleifer and B. C. Amick, “System repines time and method of pay: Stress effects of computer-based tasks.” International Journal of Human-Computer Interaction, vol.1 No.1, pp. 23-39, 1989

202. A. Sears and M. S. Borella, “WANDS: Tools for designing and testing distributed documents.” Technical Report #97-01, School of Computer Science, DePaul University, Chicago, IL, 1997.

203. P. Sevcik, “understanding how users view application performance” , Business Communications Review Magazine, Jul 19,2002.

204. R. B. Miller, “Response time in man-computer conversational transactions.”, Proc. AFIPS Fall Joint Computer Conference Vol. 33, Montvale, NJ, pp. 267-277, 1968

205. N. Bhatti, A. Bouch, A. Kuchinsky, "Integrating User-Perceived Quality into Web Server Design,", Proc. of 9th International World Wide Web Conference, Amesterdam , The Netherlands, pp. 1-26, May 2000

206. Zona Research, Inc."The Need for Speed", July 1999 and “The Need for Speed II.” Zona Market Bulletin, (5), April 2001

207. B. A. Myers, “The Importance of Percent-Done Progress Indicators for Computer-Human Interfaces.” Proceedings of CHI’85, San Francisco, CA, April 1985.

208. J. Ramsay, A. Barbasi, J. Preece,"A psychological investigation of long retrieval times on the World Wide Web," Interacting With Computers, Elsevier, March 1998.

209. P. Selvidge, "How Long is Too Long to Wait for a Website to Load?" , Usability News, Wichita State University, July 1999.

210. J. Nielsen, “Usability Engineering”, San Francisco: Morgan Kaufmann, 1994.

211. Accenture: Global management consulting and technology services company [online]. Available: http://www.accenture.com

212. J. Klein, “Computer Response to user frustration.”, MIT Master Thesis, September 1998.

213. J. Klein, Y. Moon, R. W. Picard, ”this computer responds to user frustration”, Conference on Human Factors in Computing Systems CHI 99, Pennsylvania, USA, 1999

214. K. Kreijns, P. A. Kirschner and W. Jochems. “The sociability of computer-supported collaborative learning environments.”, Journal of Education Technology & Society, vol. 5 no. 1, pp. 8–22. Retrieved February 2, 2003

215. Nokia, “Quality of Experience (QoE) of mobile services: Can it be measured and improved?”, Whitepaper, 2004

216. A. Bouch, A. Kuchinsky, N. Bhatti, “Quality is in the Eye of the Beholder: Meeting Users’ Requirements for Internet Quality of Service”, HP labs technical report HPL-2000-4, January 2000

217. L. Yu, Y. Min-hua and Z. Hui-min; “The handoff schemes in mobile IP,” Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual vol. 1, pp. 485 - 489 , 22-25 April 2003

218. R. Braden ,“Requirements for Internet Hosts - Communication Layers”, IETF RFC 1122 , October 1989

219. F. Chai, “Micro-Mobility Management and Freeze-TCP Implementation in a Wireless Network”, ME THESIS , Electrical & Computer Engineering Dept. University of Canterbury, New Zealand.

Appendix A VNC in Multi-environment mode on UNIX

To start up multiple vncservers at boot time on Solaris the following entries were made in the configuration files to set up the ports for vnc services. Different graphics window sizes are used[1].

First you should create multiple shell scripts in /usr/local/bin where Xvnc is installed and you can call them /usr/local/bin/Xvncstart1 and /Xvncstart2 and so on and the window sizes can be changed between one file and another.

```
#!bin/sh

/usr/local/bin/Xvnc –inetd –query localhost –once –localhost –qeometry 800x600
```

The following must be added to /etc/inetd.conf:

```
vnc1-i stream tcp nowait nobody /usr/local/bin/tcpd /usr/local/bin/Xvncstart1

vnc2-i stream tcp nowait nobody /usr/local/bin/tcpd /usr/local/bin/Xvncstart2

vnc3-i stream tcp nowait nobody /usr/local/bin/tcpd /usr/local/bin/Xvncstart3
```

Add the corresponding ports to /etc/services

```
vnc1-i  5901/tcp                #VNC via inetd (800x600)

vnc2-i  5902/tcp                #VNC via inetd (1024x768)

vnc3-i  5903/tcp                #VNC via inetd (12804x1024)
```

Then you either restart inetd or send a hangup to inetd

```
ps –ef | grep inetd (to find inetd's PID)

kill –HUP PID
```

To avoid editing inetd.conf the following procedure[2] can replace the previous one:

First become a root. Then create a file called /etc/init.d/rc.vnc with the following contents:

PT[1] TPThanks for Peter Rotheroe from the Realvnc mailing list for his assistance

[2] thank you for Mike Miller in RealVNC mailing list for his contribution

```
------------------------------start file on next line--------------------------------------------

#!/bin/sh
# Startup/Stop script for vncservers for some users.
#

case "$1" in

'start')
  /bin/su - bob -c "/usr/local/bin/vncserver :1"
  /bin/su - sally -c "/usr/local/bin/vncserver :2"
  /bin/su - jim -c "/usr/local/bin/vncserver :3"
  ;;

'stop')
  /bin/su - bob -c "/usr/local/bin/vncserver -kill :1"
  /bin/su - sally -c "/usr/local/bin/vncserver -kill :2"
  /bin/su - jim -c "/usr/local/bin/vncserver -kill :3"
  ;;

*)
  echo "Usage: /etc/init.d/rc.vnc { start | stop }"
  ;;

esac
------------------------------end file on previous line----------------------------------------
```

The names bob, sally and jim are usernames. Modify accordingly. The corresponding numbers :1, :2, :3 for VNC may also need to be changed.

Make sure that /etc/init.d/rc.vnc has the correct ownership/permissions:

```
# chown root:other /etc/init.d/rc.vnc
# chmod 744 /etc/init.d/rc.vnc
```

Next create symbolic links to that file as follows:

```
# cd /etc/rc0.d
# ln -s ../init.d/rc.vnc K40rc.vnc
# cd /etc/rc2.d
# ln -s ../init.d/rc.vnc S99rc.vnc
# cd /etc/rc2.d
# ln -s ../init.d/rc.vnc S99rc.vnc
```

Appendix B VNC prototype for Administration and videoconferencing

```
Import java.net.ServerSocket;

import java.net.Socket;

import java.io.IOException;

import java.util.Date;

public class CatchIP {

        public static void main(final String args[]) throws IOException {

        final ServerSocket server = new ServerSocket (5900);

while (true) {

  try {

        final Socket socket = server.accept();

        System.out.println("Connection attempted from"
+socket.getInetAddress().toString() +"at"+new Date());

        socket.close();

        }

        catch (final IOException e) {

        System.out.println("Exception : "+ e);

}}}}
```

Program 1 IP catch program

```
<html>
<head>
 <TITLE>Browser Capability</TITLE>
 <HTA:APPLICATION ID="HTAEx"
 APPLICATIONNAME="HTAEx"
 ICON="e.ico"
 WINDOWSTATE="normal">
</head>

<body>

<span id=AddressBar style="overflow: none">
<span id=AddText>
<b><font color=”blue” face=”verdana” size=”2”> Connect to
Address</font></b></span>

<input type=text value=http://www.city.ac.uk id=TheAddress
 style="width: expression(document.body.clientWidth -
AddText.offsetWidth - AddGo.offsetWidth - 45)">

<input type=button value="Go" id=AddGo onclick="navigate()"><br>
<span>
<br>
<iframe src="http://www.city.ac.uk" id=TheFrame
style="width:100%; height: 85%"></iframe>

<script language=JScript>
function navigate() {
 document.all.TheFrame.src = TheAddress.value;
}

function clickShortcut() {
 if (window.event.keyCode == 13) {
 navigate()
 }
}
TheAddress.onkeypress =
clickShortcut;
</script>
</body>
</html>
```

Program 2 browser.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HEAD><TITLE>Two VNC Desktops</TITLE>
<HTA:APPLICATION
     ID = "vnc"
     APPLICATIONNAME = "VNC"
     BORDER= "thick"
     BORDERSTYLE = "complex"
     CAPTION = "YES"
```

```
    MAXIMIZEBUTTON = "YES"
    MINIMIZEBUTTON = "YES"
  SHOWINTASKBAR = "YES"
  SINGLEINSTANCE = "NO"
    SYSMENU = "YES"
  VERSION = "1.095"
  WINDOWSTATE = "maximize"
>
</HEAD>
<FRAMESET cols=1*,1*>
<FRAME src="browser.html">
<FRAME src="browser2.html">
</FRAMESET>
</HTML>
```

Program 3 VNC.hta

```
<html>
<head>
<title> multiple vnc</title>
<style>
.clsButton {font-family: Arial;
   font-size: 10pt;
   background-color: #E0EDFF;
   color: #0000FF;
}</style>
<HTA:APPLICATION
  ID = "myVNC"
    APPLICATIONNAME = "mVNC"
  BORDER= "thick"
    BORDERSTYLE = "complex"
    CAPTION = "YES"
    MAXIMIZEBUTTON = "YES"
    MINIMIZEBUTTON = "YES"
  SHOWINTASKBAR = "YES"
  SINGLEINSTANCE = "YES"
    SYSMENU = "YES"
  VERSION = "1.095"
  WINDOWSTATE = "maximize"
></HEAD>
<body>
<IFRAME application="yes" SRC=" browser.html" WIDTH="1000"
height="900"></iframe>
<IFRAME application="yes" SRC=" browser2.html" WIDTH="1000"
height="900"></iframe>
```

```
<IFRAME application="yes" SRC="browser3.html" WIDTH="1000"
height="900"></iframe>
</body>
</HTML>
```

Program 4 horizontalVNC

```
<!--
This file demonstrates using VNCX from a webpage.  It is provided as is.
This code is provided as is by Thong Nguyen (tummy@technologist.com).
You can use this code anyway you like as long as you provide a link to
http://tummy.veridicus.com/tummy/programming/vncx.
Please also include this comment block.
-->
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 5.0">
<TITLE></TITLE>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT="Error(number,
message, variant)">
<!--
 VNCViewer1_Error(number, message, variant)
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT="Warning(number,
message, variant)">
<!--
 VNCViewer1_Warning(number, message, variant)
//-->
</SCRIPT>
<meta name="Microsoft Border" content="none">
</HEAD>
<BODY LANGUAGE=javascript onload="return window_onload()">

<P>
<TABLE border=0 cellPadding=1 cellSpacing=1 style="WIDTH: 100%" width="100%"
height="100%">

  <TR>
    <TD align=left height=110 style="HEIGHT: 20px" vAlign=center>
      <TABLE border=1 cellPadding=8 cellSpacing=0 width="100%" background=""
borderColor=blue borderColorDark=black
      borderColorLight=green height=1
      style="HEIGHT: 1px; WIDTH: 100%">
```

```
     <TR>
      <TD align=left vAlign=center width="80%">
       <p align="center">
        <INPUT id=buttonConnect name=buttonConnect type=submit
value=Connect LANGUAGE=javascript onclick="return buttonConnect_onclick()"
style="BACKGROUND-COLOR: white; FONT-WEIGHT: bold; LEFT: 8px; TOP:
8px">
        &nbsp;&nbsp;
        <INPUT id=textServer name=server value=""
       style="HEIGHT: 22px; WIDTH: 105px" size="20"
       >
        &nbsp;&nbsp;&nbsp;
        <INPUT id=buttonDisconnect name=buttonDisconnect type=button
value=Disconnect LANGUAGE=javascript onclick="return
buttonDisconnect_onclick()" style="BACKGROUND-COLOR: white; FONT-
WEIGHT: bold">
        &nbsp;&nbsp; </p>
       <p align="center">
        <INPUT id=checkSendMouse name=checkbox1 onpropertychange="return
checkSendMouse_onpropertychange()" type =checkbox LANGUAGE=javascript
CHECKED value="ON">
        &nbsp;<FONT
       face=Arial>SendMouse&nbsp;&nbsp;
        <INPUT id=checkSendKeys
       name=checkbox2 onpropertychange="return
checkSendKeys_onpropertychange()" type
       =checkbox LANGUAGE=javascript CHECKED value="ON"
       >
        &nbsp; SendKeys</FONT></p>
      </TD>
      <TD align=right vAlign=center width="20%"><STRONG><FONT
face=Arial>Zoom&nbsp;&nbsp;<STRONG><FONT face=Arial>
<SELECT
       id=zoom name="select1" onpropertychange="return
zoom_onpropertychange()" style  ="HEIGHT: 22px; LEFT: 127px; TOP: 8px; WIDTH:
76px"
       LANGUAGE="javascript"> <OPTION
        value=10>10%</OPTION><OPTION value=20>20%</OPTION><OPTION
value=30>30%</OPTION><OPTION
        value=40>40%</OPTION><OPTION value=50>50%</OPTION><OPTION
        value=60>60%</OPTION><OPTION value=70>70%</OPTION><OPTION
        value=80>80%</OPTION><OPTION value=90>90%</OPTION><OPTION
        selected value=100>100%</OPTION><OPTION
value=110>110%</OPTION><OPTION
        value=120>120%</OPTION><OPTION
value=130>130%</OPTION><OPTION
```

```
        value=140>140%</OPTION><OPTION
value=150>150%</OPTION><OPTION
        value=160>160%</OPTION><OPTION
value=170>170%</OPTION><OPTION
        value=180>180%</OPTION><OPTION
value=190>190%</OPTION><OPTION
        value=200>200%</OPTION><OPTION

value=""></OPTION></SELECT></FONT></STRONG></FONT></STRONG></TD></
TR></TABLE></TD></TR>

 <TR>
   <TD align=middle vAlign=center>
     <div align="center"><OBJECT classid=clsid:EFAB8D1F-794A-4C47-B834-
53653E05A441
     id=VNCViewer1
style="HEIGHT: 311px; WIDTH: 341px" width="341" height="311">
      </OBJECT> </div>
   </TD></TR></TABLE>

<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
<!--

/**
 * Some constants from the VNCXLib typelibrary.
 */

// VNCXScrollbarsEnum
var vncxScrollbarNone = 0;
var vncxScrollbarHorizontal = 1;
var vncxScrollbarVertical = 2;
var vncxScrollbarBoth = 3;

// VNCXWarningsEnum
var vncxWarningHandshakeAuthCancelled = 0x2ff;

function VNCViewer1_Connected(ServerName)
{
        clear();
        VNCViewer1.Start();

        window.status = "Connected to " + VNCViewer1.ServerAddress + "  ::
ServerVersion = " + VNCViewer1.ServerVersion;
        window.document.title = ServerName;

        zoom_onpropertychange();
```

```
}

function setSize()
{
        if (VNCViewer1.State > 0)
        {
                VNCViewer1.style.width = VNCViewer1.ViewWidth;
                VNCViewer1.style.height = VNCViewer1.ViewHeight;
        }
}

function VNCViewer1_Password()
{
        var vempty;

        VNCViewer1.Password = window.showModalDialog("vncx_password.htm",
vempty, "dialogWidth:262pt;dialogHeight=80pt");
}

function VNCViewer1_Disconnected()
{
        window.status = "Disconnected";
}

function buttonConnect_onclick()
{
        var vempty;

        checkSendKeys_onpropertychange();
        checkSendMouse_onpropertychange();

        clear();
        save();

        //VNCViewer1.ClearProps();
        VNCViewer1.Alignment = 16;
        VNCViewer1.Stretch = true;
        VNCViewer1.Display = 0; // default display
        VNCViewer1.Connect(textServer.value, vempty, vempty, vempty);
}


function zoom_onpropertychange()
{
        VNCViewer1.Scrollbars = vncxScrollbarNone;
```

```
    if (zoom.value == 100)
    {
        VNCViewer1.Stretch = false;
    }
    else
    {
        VNCViewer1.Stretch = true;
    }

    VNCViewer1.StretchX(zoom.value / 10, 10);
    VNCViewer1.StretchY(zoom.value / 10, 10);

    setSize();
}

function VNCViewer1_Error(number, message, variant)
{
    window.status = "Error";
    clear();
    alert("Error " + (number) + ": " + message);
}

function VNCViewer1_Warning(number, message, variant)
{
    clear();
    window.status = ("Warning " + (number) + ": " + message);
}

function checkSendKeys_onpropertychange()
{
    VNCViewer1.RelayKeys = checkSendKeys.checked;
}

function checkSendMouse_onpropertychange()
{
    VNCViewer1.RelayMouse = checkSendMouse.checked;
}

function clear()
{
    VNCViewer1.Password = "";
}

function buttonDisconnect_onclick()
{
    VNCViewer1.Stop();
```

```
}

function window_onload()
{
        // load settings
        load();
}

// Save settings to the cookie.
function save()
{
        var date = new Date();

        // cookie valid for 2 months
        date.setMonth(date.getMonth() + 2);

        setCookie("server", textServer.value, date, null, null, null);
        setCookie("sendkeys", checkSendKeys.checked, date, null, null, null);
        setCookie("sendmouse", checkSendMouse.checked.value, date, null, null,
null);
        setCookie("zoom", zoom.value, date, null, null, null);
}

// Load settings from the cookie.
function load()
{
        var v;

        v = getCookie("server");
        if (v) textServer.value = v;

        v = getCookie("sendkeys");
        if (v) checkSendKeys.value = v;

        v = getCookie("sendmouse");
        if (v) checkSendMouse.value = v;

        v = getCookie("zoom");
        if (v)
        {
                zoom.value = v;
        }
}

function setCookie(name, value, expires, path, domain, secure)
{
  var curCookie = name + "=" + escape(value) +
      ((expires) ? "; expires=" + expires.toGMTString() : "") +
      ((path) ? "; path=" + path : "") +
      ((domain) ? "; domain=" + domain : "") +
      ((secure) ? "; secure" : "");

  document.cookie = curCookie;
}

function getCookie(name)
{
  var dc = document.cookie;
  var prefix = name + "=";
  var begin = dc.indexOf("; " + prefix);

  if (begin == -1)
  {
    begin = dc.indexOf(prefix);
    if (begin != 0) return null;
  }
  else
  {
    begin += 2;
  }

  var end = document.cookie.indexOf(";", begin);

  if (end == -1)
  {
    end = dc.length;
   }

  return unescape(dc.substring(begin + prefix.length, end));
}

function deleteCookie(name, path, domain)
{
  if (getCookie(name))
  {
    document.cookie = name + "=" +
    ((path) ? "; path=" + path : "") +
    ((domain) ? "; domain=" + domain : "") +
    "; expires=Thu, 01-Jan-70 00:00:01 GMT";
  }
}
```

```
function fixDate(date)
{
 var base = new Date(0);
 var skew = base.getTime();

 if (skew && 0)
 {
  date.setTime(date.getTime() - skew);
 }
}



//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1
EVENT=Connected(ServerName)>
<!--
 VNCViewer1_Connected(ServerName)
//-->
</SCRIPT>
<SCRIPT returnvalue="true" LANGUAGE=javascript FOR=VNCViewer1
EVENT="Password">
<!--
        VNCViewer1_Password();
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT=Disconnected>
<!--
 VNCViewer1_Disconnected()
//-->
</SCRIPT>
<p>&nbsp;</p>

</BODY>
</HTML>
```

Program 5 VNCX-ActiveX window

```
<!--
This file demonstrates using VNCX from a webpage.  It is provided as is.
This code is provided as is by Thong Nguyen (tummy@technologist.com).
You can use this code anyway you like as long as you provide a link to
http://tummy.veridicus.com/tummy/programming/vncx.
Please also include this comment block.
-->
<HTML>
```

```
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 5.0">
<TITLE></TITLE>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT="Error(number,
message, variant)">
<!--
 VNCViewer1_Error(number, message, variant)
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT="Warning(number,
message, variant)">
<!--
 VNCViewer1_Warning(number, message, variant)
//-->
</SCRIPT>
<meta name="Microsoft Border" content="none">
</HEAD>
<BODY LANGUAGE=javascript onload="return window_onload()">

<P>
<TABLE border=0 cellPadding=1 cellSpacing=1 style="WIDTH: 100%" width="100%"
height="100%">

  <TR>
    <TD align=left height=20 style="HEIGHT: 20px" vAlign=center>
      <TABLE border=1 cellPadding=8 cellSpacing=0 width="100%" background=""
borderColor=blue borderColorDark=black
      borderColorLight=green height=1
      style="HEIGHT: 1px; WIDTH: 100%">

        <TR>
          <TD align=left vAlign=center><INPUT id=buttonConnect name=buttonConnect
type=submit value=Connect LANGUAGE=javascript onclick="return
buttonConnect_onclick()" style="BACKGROUND-COLOR: white; FONT-WEIGHT:
bold; LEFT: 8px; TOP: 8px">&nbsp;&nbsp;
           <INPUT id=textServer name=server value=""
          style="HEIGHT: 22px; WIDTH: 105px" size="20"
          >&nbsp;&nbsp;&nbsp;&nbsp; <INPUT id=buttonDisconnect
name=buttonDisconnect type=button value=Disconnect LANGUAGE=javascript
onclick="return buttonDisconnect_onclick()" style="BACKGROUND-COLOR: white;
FONT-WEIGHT: bold">&nbsp;&nbsp;
           <INPUT id=checkSendMouse name=checkbox1 onpropertychange="return
checkSendMouse_onpropertychange()" type =checkbox LANGUAGE=javascript
CHECKED value="ON">&nbsp;<FONT
           face=Arial>SendMouse&nbsp;&nbsp;
          <INPUT id=checkSendKeys
          name=checkbox2 onpropertychange="return
checkSendKeys_onpropertychange()" type
          =checkbox LANGUAGE=javascript CHECKED value="ON"
         >&nbsp; SendKeys</FONT></TD>
        <TD align=right vAlign=center><STRONG><FONT
face=Arial>Zoom&nbsp;&nbsp;<STRONG><FONT face=Arial><SELECT
          id=zoom name="select1" onpropertychange="return
zoom_onpropertychange()" style  ="HEIGHT: 22px; LEFT: 127px; TOP: 8px; WIDTH:
76px"
          LANGUAGE="javascript"> <OPTION
            value=10>10%</OPTION><OPTION value=20>20%</OPTION><OPTION
value=30>30%</OPTION><OPTION
            value=40>40%</OPTION><OPTION value=50>50%</OPTION><OPTION
            value=60>60%</OPTION><OPTION value=70>70%</OPTION><OPTION
            value=80>80%</OPTION><OPTION value=90>90%</OPTION><OPTION
            selected value=100>100%</OPTION><OPTION
value=110>110%</OPTION><OPTION
            value=120>120%</OPTION><OPTION
value=130>130%</OPTION><OPTION
            value=140>140%</OPTION><OPTION
value=150>150%</OPTION><OPTION
            value=160>160%</OPTION><OPTION
value=170>170%</OPTION><OPTION
            value=180>180%</OPTION><OPTION
value=190>190%</OPTION><OPTION
            value=200>200%</OPTION><OPTION

value=""></OPTION></SELECT></FONT></STRONG></FONT></STRONG></TD></
TR></TABLE></TD></TR>

  <TR>
    <TD align=middle vAlign=center>
      <OBJECT classid=clsid:EFAB8D1F-794A-4C47-B834-53653E05A441
      id=VNCViewer1
style="HEIGHT: 311px; WIDTH: 341px" width="341" height="311"></OBJECT>
</TD></TR></TABLE>

<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
<!--

/**
 * Some constants from the VNCXLib typelibrary.
 */

// VNCXScrollbarsEnum
var vncxScrollbarNone = 0;
var vncxScrollbarHorizontal = 1;
var vncxScrollbarVertical = 2;
var vncxScrollbarBoth = 3;

// VNCXWarningsEnum
var vncxWarningHandshakeAuthCancelled = 0x2ff;

function VNCViewer1_Connected(ServerName)
{
        clear();
        VNCViewer1.Start();

        window.status = "Connected to " + VNCViewer1.ServerAddress + "  ::
ServerVersion = " + VNCViewer1.ServerVersion;
        window.document.title = ServerName;

        zoom_onpropertychange();
}

function setSize()
{
        if (VNCViewer1.State > 0)
        {
                VNCViewer1.style.width = VNCViewer1.ViewWidth;
                VNCViewer1.style.height = VNCViewer1.ViewHeight;
        }
}

function VNCViewer1_Password()
{
        var vempty;

        VNCViewer1.Password = window.showModalDialog("vncx_password.htm",
vempty, "dialogWidth:262pt;dialogHeight=80pt");
}

function VNCViewer1_Disconnected()
{
        window.status = "Disconnected";
}

function buttonConnect_onclick()
{
        var vempty;

        checkSendKeys_onpropertychange();
```

```
    checkSendMouse_onpropertychange();

    clear();
    save();

    //VNCViewer1.ClearProps();
    VNCViewer1.Alignment = 16;
    VNCViewer1.Stretch = true;
    VNCViewer1.Display = 0; // default display
    VNCViewer1.Connect(textServer.value, vempty, vempty, vempty);
}


function zoom_onpropertychange()
{
    VNCViewer1.Scrollbars = vncxScrollbarNone;

    if (zoom.value == 100)
    {
        VNCViewer1.Stretch = false;
    }
    else
    {
        VNCViewer1.Stretch = true;
    }

    VNCViewer1.StretchX(zoom.value / 10, 10);
    VNCViewer1.StretchY(zoom.value / 10, 10);

    setSize();
}

function VNCViewer1_Error(number, message, variant)
{
    window.status = "Error";
    clear();
    alert("Error " + (number) + ": " + message);
}

function VNCViewer1_Warning(number, message, variant)
{
    clear();
    window.status = ("Warning " + (number) + ": " + message);
}

function checkSendKeys_onpropertychange()
```

```
{
        VNCViewer1.RelayKeys = checkSendKeys.checked;
}

function checkSendMouse_onpropertychange()
{
        VNCViewer1.RelayMouse = checkSendMouse.checked;
}

function clear()
{
        VNCViewer1.Password = "";
}

function buttonDisconnect_onclick()
{
        VNCViewer1.Stop();
}

function window_onload()
{
        // load settings
        load();
}

// Save settings to the cookie.
function save()
{
        var date = new Date();

        // cookie valid for 2 months
        date.setMonth(date.getMonth() + 2);

        setCookie("server", textServer.value, date, null, null, null);
        setCookie("sendkeys", checkSendKeys.checked, date, null, null, null);
        setCookie("sendmouse", checkSendMouse.checked.value, date, null, null,
null);
        setCookie("zoom", zoom.value, date, null, null, null);
}

// Load settings from the cookie.
function load()
{
        var v;

        v = getCookie("server");
```

```
    if (v) textServer.value = v;

    v = getCookie("sendkeys");
    if (v) checkSendKeys.value = v;

    v = getCookie("sendmouse");
    if (v) checkSendMouse.value = v;

    v = getCookie("zoom");
    if (v)
    {
        zoom.value = v;
    }
}

function setCookie(name, value, expires, path, domain, secure)
{
 var curCookie = name + "=" + escape(value) +
   ((expires) ? "; expires=" + expires.toGMTString() : "") +
   ((path) ? "; path=" + path : "") +
   ((domain) ? "; domain=" + domain : "") +
   ((secure) ? "; secure" : "");

 document.cookie = curCookie;
}

function getCookie(name)
{
 var dc = document.cookie;
 var prefix = name + "=";
 var begin = dc.indexOf("; " + prefix);

 if (begin == -1)
 {
  begin = dc.indexOf(prefix);
  if (begin != 0) return null;
 }
 else
 {
  begin += 2;
 }

 var end = document.cookie.indexOf(";", begin);

 if (end == -1)
 {
    end = dc.length;
   }

  return unescape(dc.substring(begin + prefix.length, end));
}

function deleteCookie(name, path, domain)
{
  if (getCookie(name))
  {
    document.cookie = name + "=" +
    ((path) ? "; path=" + path : "") +
    ((domain) ? "; domain=" + domain : "") +
    "; expires=Thu, 01-Jan-70 00:00:01 GMT";
  }
}

function fixDate(date)
{
  var base = new Date(0);
  var skew = base.getTime();

  if (skew && 0)
  {
    date.setTime(date.getTime() - skew);
  }
}


//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1
EVENT=Connected(ServerName)>
<!--
 VNCViewer1_Connected(ServerName)
//-->
</SCRIPT>
<SCRIPT returnvalue="true" LANGUAGE=javascript FOR=VNCViewer1
EVENT="Password">
<!--
		VNCViewer1_Password();
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT=Disconnected>
<!--
 VNCViewer1_Disconnected()
```

```
//-->
</SCRIPT>
<p>&nbsp;</p>

</BODY>
</HTML>
```

Program 6 VNCX-ie.htm

```
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 5.0">
<TITLE>Password</TITLE>
<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
<!--

function window_onunload()
{
        window.returnValue = password.value;
}

function buttonok_onclick()
{
        window.close();
}

function password_onkeyup()
{

}

//-->
</SCRIPT>
<meta name="Microsoft Border" content="none">
</HEAD>
<BODY LANGUAGE=javascript onunload="return window_onunload()">

<P>
<TABLE border=0 cellPadding=1 cellSpacing=1 style="HEIGHT: 56px; WIDTH:
343px"
width="75%">

 <TR>
  <TD><STRONG><FONT face=Arial>PASSWORD</FONT></STRONG></TD>
  <TD>
```

```
   <INPUT id=password name=password type=password LANGUAGE=javascript
onkeyup="return password_onkeyup()" size="20"></TD>
   <TD><INPUT id=buttonok name=buttonok style="HEIGHT: 24px; WIDTH: 65px"
type=submit value=Ok LANGUAGE=javascript onclick="return
buttonok_onclick()"></TD></TR>
  <TR>
   <TD></TD>
   <TD colSpan=2></TD></TR></TABLE></P>

</BODY>
</HTML>
```

Program 7 VNCX-password.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HEAD><TITLE>Two VNC Desktops</TITLE>

<HTA:APPLICATION

ID = "vnc"
APPLICATIONNAME = "VNC"
BORDER= "thick"
BORDERSTYLE = "complex"
CAPTION = "YES"
MAXIMIZEBUTTON = "YES"
MINIMIZEBUTTON = "YES"
SHOWINTASKBAR = "YES"
SINGLEINSTANCE = "NO"
SYSMENU = "YES"
VERSION = "1.095"
WINDOWSTATE = "maximize"
>
</HEAD>

<frameset cols="1*, 1*" scrolling= YES>
   <frame name="leftFrame" scrolling="yes" src="vncx1.htm">
   <frame name="mainFrame" src="vncx2.htm">
  </frameset>
</HTML>
```

Program 8 Two ActiveX Windows

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;



public class multiVNC {

// Display the desktop in a top-level frame
JFrame frame = new JFrame("multi-VNC by Dana Al-Malki, City University UK.");
JDesktopPane desktop = new JDesktopPane();
Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
String HostIPNumber;
String PortNumber;

static final int width = 512;
static final int height = 384;


public static void main(String[] args)
{ new multiVNC();
}


public multiVNC()
{
CreateMenu();

frame.getContentPane().add(desktop, BorderLayout.CENTER);
desktop.setBackground(new Color(0,60,40));

frame.getContentPane().add(desktop, BorderLayout.CENTER);
frame.setSize(dim);
frame.setVisible(true);

}

public void CreateMenu()
{
// Create the menu bar
JMenuBar menuBar = new JMenuBar();
// Create a menu
JMenu menu = new JMenu("File");
menuBar.add(menu);
```

```
JMenuItem newItem = new JMenuItem("New VNC Window");
newItem.addActionListener(new java.awt.event.ActionListener()
{ public void actionPerformed(java.awt.event.ActionEvent e)
{
  boolean dataGot = false;
  while(dataGot != true)
  {
  HostIPNumber  = JOptionPane.showInputDialog( "Enter Host IP" );
  PortNumber  = JOptionPane.showInputDialog( "Enter Port" );
   dataGot = true;
  }
  System.out.println("goin to generate frame");

  CreateNewInternalFrame(HostIPNumber,PortNumber);
}
});
JMenuItem screenshotItem = new JMenuItem("Take Screenshot");
screenshotItem.addActionListener(new java.awt.event.ActionListener()
{ public void actionPerformed(java.awt.event.ActionEvent e)
{ //CreateScreenshot();
}
});

JMenuItem exitItem = new JMenuItem("Exit");
screenshotItem.addActionListener(new java.awt.event.ActionListener()
{ public void actionPerformed(java.awt.event.ActionEvent e)
{ System.exit(1);
}
});


menu.add(newItem);
// Install the menu bar in the frame
frame.setJMenuBar(menuBar);


}

public void CreateNewInternalFrame(String hip,String hop)
{

 String[] argv = {"HOST",hip,"PORT", hop};
```

```
  System.out.println("Going to connect to " + argv[0] + " " + argv[1] + " " + argv[2] + " "
+ argv[3]);
  vncviewer v = new vncviewer();

   v.mainArgs = argv;
   v.inAnApplet = false;

  // v.f.add("Center", v);


// Create an internal frame
boolean resizable = true;
boolean closeable = true;
boolean maximizable = true;
boolean iconifiable = true;
String title = hip + ":" + hop;    //JInternalFrame iframe

 v.jif = new JInternalFrame(title, resizable, closeable,
maximizable, iconifiable);



// Set an initial size

v.jif.setSize(width, height);

// Display the internal frame
v.jif.setVisible(true);
setInternalFrameToCenter(v.jif);
// Add components to internal frame...


v.jif.getContentPane().add(v);
desktop.add(v.jif);
v.init();
v.start();
System.out.println("started...");
}



public void setInternalFrameToCenter(JInternalFrame jiffy)
{
// Determine the new location of the window
```

```
int w = jiffy.getSize().width;
int h = jiffy.getSize().height;
int x = (dim.width-w)/2;
int y = (dim.height-h)/2;
// Move the window
jiffy.setLocation(x, y);

}

}
```

Program 9 MultiVNC

```
//
//  Copyright (C) 1999 AT&T Laboratories Cambridge.  All Rights Reserved.
//
//  This is free software; you can redistribute it and/or modify
//  it under the terms of the GNU General Public License as published by
//  the Free Software Foundation; either version 2 of the License, or
//  (at your option) any later version.
//
//  This software is distributed in the hope that it will be useful,
//  but WITHOUT ANY WARRANTY; without even the implied warranty of
//  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
//  GNU General Public License for more details.
//
//  You should have received a copy of the GNU General Public License
//  along with this software; if not, write to the Free Software
//  Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA  02111-1307,
//  USA.
//

//
// vncviewer.java - the VNC viewer applet.  This class mainly just sets up the
// user interface, leaving it to the vncCanvas to do the actual rendering of
// a VNC desktop.
//

import java.awt.*;
import java.io.*;
import javax.swing.*;             *********************

public class vncviewer extends java.applet.Applet
                         implements java.lang.Runnable
{
 boolean inAnApplet = true;

 //
 // main() is called when run as a java program from the command line.  It

 // simply creates a frame and runs the applet inside it.
 //


 JInternalFrame jif;              **********************
 String[] mainArgs;
 String host;
 int port;
 rfbProto rfb;
 Thread rfbThread;
//GridBagLayout gridbag;
 JPanel buttonPanel;
 JButton disconnectButton;
 JButton optionsButton;
 JButton clipboardButton;
 JButton ctrlAltDelButton;
 optionsFrame options;
 clipboardFrame clipboard;
 authenticationPanel authenticator;

 public static void main(String[] argv) {
   vncviewer v = new vncviewer();
   v.mainArgs = argv;
   v.inAnApplet = false;

  // v.f = new Frame("VNC");
   //v.f.add("Center", v);

   v.init();
   v.start();
 }
 //
 // init()
 //

 public void init() {

   readParameters();

   options = new optionsFrame(this);
   clipboard = new clipboardFrame(this);
   authenticator = new authenticationPanel();

   rfbThread = new Thread(this);
   rfbThread.start();
 }

 public void update(Graphics g) {
 }

 //
 // run() - executed by the rfbThread to deal with the RFB socket.
 //

 public void run() {

  // gridbag = new GridBagLayout();
   jif.getContentPane().setLayout(new FlowLayout(FlowLayout.LEFT, 0, 0));
//gridbag);                    **********************

   buttonPanel = new JPanel();
   buttonPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 0, 0));
   disconnectButton = new JButton("Disconnect");
   disconnectButton.disable();
   buttonPanel.add(disconnectButton);
   optionsButton = new JButton("Options");
   buttonPanel.add(optionsButton);
   clipboardButton = new JButton("Clipboard");
   clipboardButton.disable();
   buttonPanel.add(clipboardButton);
   ctrlAltDelButton = new JButton("Send Ctrl-Alt-Del");
   ctrlAltDelButton.disable();
   buttonPanel.add(ctrlAltDelButton);
```

```
    /*GridBagConstraints gbc = new GridBagConstraints();
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    gbc.anchor = GridBagConstraints.NORTHWEST;
    gridbag.setConstraints(buttonPanel,gbc);
    */
    jif.getContentPane().add(buttonPanel);     *******************

    try {
      connectAndAuthenticate();

      doProtocolInitialisation();

      vncCanvas vc = new vncCanvas(this);
     // gbc.weightx = 1.0;
     // gbc.weighty = 1.0;
     // gridbag.setConstraints(vc,gbc);


/*
JScrollPane scrollpane = new JScrollPane(vc,
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS );
scrollpane.setPreferredSize(new Dimension(VueDeCiel.width, VueDeCiel.height));
   add(scrollpane);

*/
 add(vc);

      if (!inAnApplet) {
            //f.setTitle(rfb.desktopName);
            //f.pack();
      } else {
            validate();
      }

      disconnectButton.enable();
      clipboardButton.enable();
      ctrlAltDelButton.enable();

      vc.processNormalProtocol();

    } catch (Exception e) {
      e.printStackTrace();
      fatalError(e.toString());
    }

   }


   //
   // Connect to the RFB server and authenticate the user.
   //

   void connectAndAuthenticate() throws IOException {

    /*GridBagConstraints gbc = new GridBagConstraints();
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    gbc.anchor = GridBagConstraints.NORTHWEST;
    gbc.weightx = 1.0;
    gbc.weighty = 1.0;
    gbc.ipadx = 100;
    gbc.ipady = 50;
    gridbag.setConstraints(authenticator,gbc);
    */
    jif.getContentPane().add(authenticator);    ********************
    validate();
    if (!inAnApplet) {
      //f.pack();
      jif.setVisible(true);      **********************
          //f.show();
    }

    boolean authenticationDone = false;

    while (!authenticationDone) {

      synchronized(authenticator) {
            try {
             authenticator.wait();
            } catch (InterruptedException e) {
            }
      }

      rfb = new rfbProto(host, port, this);

      rfb.readVersionMsg();

      System.out.println("RFB server supports protocol version " +
                            rfb.serverMajor + "." + rfb.serverMinor);

      rfb.writeVersionMsg();

      switch (rfb.readAuthScheme()) {

      case rfbProto.NoAuth:
            System.out.println("No authentication needed");
            authenticationDone = true;
            break;

      case rfbProto.VncAuth:
            byte[] challenge = new byte[16];
            rfb.is.readFully(challenge);

            String pw = authenticator.password.getText();
            if (pw.length() > 8) pw = pw.substring(0,8); // truncate to 8 chars

            if (pw.length() == 0) {
             authenticator.retry();
             break;
            }

            byte[] key = new byte[8];
            pw.getBytes(0, pw.length(), key, 0);

            for (int i = pw.length(); i < 8; i++) {
             key[i] = (byte)0;
            }

            DesCipher des = new DesCipher(key);

            des.encrypt(challenge,0,challenge,0);
            des.encrypt(challenge,8,challenge,8);

            rfb.os.write(challenge);

            int authResult = rfb.is.readInt();

            switch (authResult) {
            case rfbProto.VncAuthOK:
             System.out.println("VNC authentication succeeded");
             authenticationDone = true;
             break;
          case rfbProto.VncAuthFailed:
            System.out.println("VNC authentication failed");
            authenticator.retry();
            break;
          case rfbProto.VncAuthTooMany:
            throw new IOException("VNC authentication failed - " +
                                  "too many tries");
          default:
            throw new IOException("Unknown VNC authentication result " +
                                  authResult);
          }
          break;
    }
  }

  remove(authenticator);
}


//
// Do the rest of the protocol initialisation.
//

void doProtocolInitialisation() throws IOException {
  System.out.println("sending client init");

  rfb.writeClientInit();

  rfb.readServerInit();

  System.out.println("Desktop name is " + rfb.desktopName);
  System.out.println("Desktop size is " + rfb.framebufferWidth + " x " +
                     rfb.framebufferHeight);

  setEncodings();
}


//
// setEncodings() - send the current encodings from the options frame
// to the RFB server.
//

void setEncodings() {
  try {
    if ((rfb != null) && rfb.inNormalProtocol) {
        rfb.writeSetEncodings(options.encodings, options.nEncodings);
    }
  } catch (Exception e) {
    e.printStackTrace();
  }
}


//
// setCutText() - send the given cut text to the RFB server.
//

void setCutText(String text) {
  try {
    if ((rfb != null) && rfb.inNormalProtocol) {
        rfb.writeClientCutText(text);
    }
  } catch (Exception e) {
    e.printStackTrace();
  }
}


//
// Respond to an action i.e. button press
//

public synchronized boolean action(Event evt, Object what) {

  if (evt.target == optionsButton) {

    if (options.isVisible()) {
        options.hide();
    } else {
        options.show();
    }

  } else if (evt.target == disconnectButton) {

    System.out.println("disconnect");
    options.dispose();
    clipboard.dispose();
```

```
      if (inAnApplet) {
            removeAll();
            rfb.close();
            rfb = null;
            JLabel l = new JLabel("Disconnected");          ***********
            jif.getContentPane().setLayout(new FlowLayout(FlowLayout.LEFT, 30, 30));
                                                            ***************
            jif.getContentPane().add(l);              ***************
            jif.getContentPane().validate();                ***************
            rfbThread.stop();
      } else {
            System.exit(1);
      }

    } else if (evt.target == clipboardButton) {

      if (clipboard.isVisible()) {
            clipboard.hide();
      } else {
            clipboard.show();
      }

    } else if (evt.target == ctrlAltDelButton) {

      try {
            Event ctrlAltDelEvent = new Event(null, 0, null);

            ctrlAltDelEvent.key = 127;
            ctrlAltDelEvent.modifiers = Event.CTRL_MASK | Event.ALT_MASK;

            ctrlAltDelEvent.id = Event.KEY_PRESS;
            rfb.writeKeyEvent(ctrlAltDelEvent);

            ctrlAltDelEvent.id = Event.KEY_RELEASE;
            rfb.writeKeyEvent(ctrlAltDelEvent);
      } catch (Exception e) {
            e.printStackTrace();
      }
    }
    return false;
  }
```

```
//
// Detect when the focus goes in and out of the applet.  See
// vncCanvas.handleEvent() for details of why this is necessary.
//

boolean gotFocus = false;

public boolean gotFocus(Event evt, Object what) {
  gotFocus = true;
  return true;
}
public boolean lostFocus(Event evt, Object what) {
  gotFocus = false;
  return true;
}


//
// encryptBytes() - encrypt some bytes in memory using a password.  Note that
// the mapping from password to key must be the same as that used on the rfb
// server side.
//
// Note also that IDEA encrypts data in 8-byte blocks, so here we will ignore
// any data beyond the last 8-byte boundary leaving it to the calling
// function to pad the data appropriately.
//

void encryptBytes(byte[] bytes, String passwd) {
  byte[] key = new byte[8];
  passwd.getBytes(0, passwd.length(), key, 0);

  for (int i = passwd.length(); i < 8; i++) {
    key[i] = (byte)0;
  }

  DesCipher des = new DesCipher(key);

  des.encrypt(bytes,0,bytes,0);
  des.encrypt(bytes,8,bytes,8);
}


//
// readParameters() - read parameters from the html source or from the
// command line.  On the command line, the arguments are just a sequence of
// param_name/param_value pairs where the names and values correspond to
// those expected in the html applet tag source.
//

public void readParameters() {
 host = readParameter("HOST", !inAnApplet);
 if (host == null) {
  host = getCodeBase().getHost();
  if (host.equals("")) {
      fatalError("HOST parameter not specified");
  }
 }

 String s = readParameter("PORT", true);
 port = Integer.parseInt(s);
}

public String readParameter(String name, boolean required) {
 if (inAnApplet) {
  String s = getParameter(name);
  if ((s == null) && required) {
      fatalError(name + " parameter not specified");
  }
  return s;
 }

 for (int i = 0; i < mainArgs.length; i += 2) {
  if (mainArgs[i].equalsIgnoreCase(name)) {
      try {
       return mainArgs[i+1];
      } catch (Exception e) {
       if (required) {
        fatalError(name + " parameter not specified");
       }
       return null;
      }
  }
 }
 if (required) {
  fatalError(name + " parameter not specified");
 }
 return null;
}
```

```
 //
 // fatalError() - print out a fatal error message.
 //

 public void fatalError(String s) {
   System.out.println(s);

   if (inAnApplet) {
     removeAll();
     JLabel l = new JLabel(s);

     jif.getContentPane().setLayout(new FlowLayout(FlowLayout.LEFT, 30, 30));
     jif.getContentPane().add(l);
     jif.getContentPane().validate();
     Thread.currentThread().stop();
   } else {
     System.exit(1);
   }
 }
}
```

Program 10 VNC viewer

********** are the lines edited and changed from the original vnc viewer code.
/*   */ are the lines not used from the original code

```
<!--
This file demonstrates using VNCX from a webpage.  It is provided as is.
This code is provided as is by Thong Nguyen (tummy@technologist.com).
You can use this code anyway you like as long as you provide a link to
http://tummy.veridicus.com/tummy/programming/vncx.
Please also include this comment block.
-->
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 5.0">
<TITLE></TITLE>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT="Error(number,
message, variant)">
<!--
```

```
 VNCViewer1_Error(number, message, variant)
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT="Warning(number,
message, variant)">
<!--
 VNCViewer1_Warning(number, message, variant)
//-->
 </SCRIPT>
<SCRIPT TYPE="text/javascript">
<!--
function popup(mylink, windowname)
{
if (! window.focus)return true;
var href;
if (typeof(mylink) == 'string')
   href=mylink;
else
   href=mylink.href;
window.open(href, windowname, 'width=300,height=450,scrollbars=no');
return false;
}
//-->
</SCRIPT>


<meta name="Microsoft Border" content="none">
</HEAD>
<BODY LANGUAGE=javascript onload="return window_onload()">

<P>
<TABLE border=0 cellPadding=1 cellSpacing=1 style="WIDTH: 100%" width="100%"
height="100%">

  <TR>
    <TD align=left height=114 style="HEIGHT: 20px" vAlign=center>
      <TABLE border=1 cellPadding=8 cellSpacing=0 width="96%" background=""
borderColor=blue borderColorDark=black
      borderColorLight=green height=1
      style="HEIGHT: 1px; WIDTH: 100%">
        <TR>
          <TD align=left vAlign=center height="81" width="79%">
            <p align="center">
              <INPUT id=buttonConnect name=buttonConnect type=submit
value=Connect LANGUAGE=javascript onclick="return buttonConnect_onclick()"
```

```
style="BACKGROUND-COLOR: white; FONT-WEIGHT: bold; LEFT: 8px; TOP:
8px">
        &nbsp;&nbsp;
        <INPUT id=textServer name=server value=""
      style="HEIGHT: 22px; WIDTH: 105px" size="20"
      >
        &nbsp;&nbsp;&nbsp;&nbsp;
        <INPUT id=buttonDisconnect name=buttonDisconnect type=button
value=Disconnect LANGUAGE=javascript onclick="return
buttonDisconnect_onclick()" style="BACKGROUND-COLOR: white; FONT-
WEIGHT: bold">
        &nbsp;&nbsp; </p>

       <p align="center">
        <INPUT id=checkSendMouse name=checkbox1 onpropertychange="return
checkSendMouse_onpropertychange()" type =checkbox LANGUAGE=javascript
CHECKED value="ON">
        &nbsp;<FONT
       face=Arial>SendMouse&nbsp;&nbsp;
        <INPUT id=checkSendKeys
       name=checkbox2 onpropertychange="return
checkSendKeys_onpropertychange()" type
       =checkbox LANGUAGE=javascript CHECKED value="ON"
      >
        &nbsp; SendKeys</FONT></p>

     </TD>
     <TD align=right vAlign=center height="81" width="21%"><STRONG><FONT
face=Arial>Zoom&nbsp;&nbsp;<STRONG><FONT face=Arial>
       <SELECT
       id=zoom name="select1" onpropertychange="return
zoom_onpropertychange()" style  ="HEIGHT: 22px; LEFT: 127px; TOP: 8px; WIDTH:
76px"
       LANGUAGE="javascript"> <OPTION
        value=10>10%</OPTION><OPTION value=20>20%</OPTION><OPTION
value=30>30%</OPTION><OPTION
        value=40>40%</OPTION><OPTION value=50>50%</OPTION><OPTION
        value=60>60%</OPTION><OPTION value=70>70%</OPTION><OPTION
        value=80>80%</OPTION><OPTION value=90>90%</OPTION><OPTION
        selected value=100>100%</OPTION><OPTION
value=110>110%</OPTION><OPTION
        value=120>120%</OPTION><OPTION
value=130>130%</OPTION><OPTION
        value=140>140%</OPTION><OPTION
value=150>150%</OPTION><OPTION
```

```
        value=160>160%</OPTION><OPTION
value=170>170%</OPTION><OPTION
        value=180>180%</OPTION><OPTION
value=190>190%</OPTION><OPTION
        value=200>200%</OPTION><OPTION

value=""></OPTION></SELECT></FONT></STRONG></FONT></STRONG></TD></
TR></TABLE></TD></TR>



<A HREF="phone.html" onClick="return popup(this, 'notes')"><img
src="phoneicon.jpg" ></A><br>

<TR>
   <TD align=middle vAlign=center>
     <div align="center"><OBJECT classid=clsid:EFAB8D1F-794A-4C47-B834-
53653E05A441
     id=VNCViewer1
style="HEIGHT: 311px; WIDTH: 341px" width="341" height="311">
       </OBJECT> </div>
   </TD></TR></TABLE>

<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
<!--

/**
 * Some constants from the VNCXLib typelibrary.
 */

// VNCXScrollbarsEnum
var vncxScrollbarNone = 0;
var vncxScrollbarHorizontal = 1;
var vncxScrollbarVertical = 2;
var vncxScrollbarBoth = 3;

// VNCXWarningsEnum
var vncxWarningHandshakeAuthCancelled = 0x2ff;

function VNCViewer1_Connected(ServerName)
{
        clear();
        VNCViewer1.Start();

        window.status = "Connected to " + VNCViewer1.ServerAddress + "  ::
ServerVersion = " + VNCViewer1.ServerVersion;
```

```
        window.document.title = ServerName;

        zoom_onpropertychange();
}

function setSize()
{
        if (VNCViewer1.State > 0)
        {
                VNCViewer1.style.width = VNCViewer1.ViewWidth;
                VNCViewer1.style.height = VNCViewer1.ViewHeight;
        }
}

function VNCViewer1_Password()
{
        var vempty;

        VNCViewer1.Password = window.showModalDialog("vncx_password.htm",
vempty, "dialogWidth:262pt;dialogHeight=80pt");
}

function VNCViewer1_Disconnected()
{
        window.status = "Disconnected";
}

function buttonConnect_onclick()
{
        var vempty;

        checkSendKeys_onpropertychange();
        checkSendMouse_onpropertychange();

        clear();
        save();

        //VNCViewer1.ClearProps();
        VNCViewer1.Alignment = 16;
        VNCViewer1.Stretch = true;
        VNCViewer1.Display = 0; // default display
        VNCViewer1.Connect(textServer.value, vempty, vempty, vempty);
}


function zoom_onpropertychange()
```

```
{
    VNCViewer1.Scrollbars = vncxScrollbarNone;

    if (zoom.value == 100)
    {
        VNCViewer1.Stretch = false;
    }
    else
    {
        VNCViewer1.Stretch = true;
    }

    VNCViewer1.StretchX(zoom.value / 10, 10);
    VNCViewer1.StretchY(zoom.value / 10, 10);

    setSize();
}function VNCViewer1_Error(number, message, variant)
{
    window.status = "Error";
    clear();
    alert("Error " + (number) + ": " + message);
}

function VNCViewer1_Warning(number, message, variant)
{
    clear();
    window.status = ("Warning " + (number) + ": " + message);
}

function checkSendKeys_onpropertychange()
{
    VNCViewer1.RelayKeys = checkSendKeys.checked;
}

function checkSendMouse_onpropertychange()
{
    VNCViewer1.RelayMouse = checkSendMouse.checked;
}

function clear()
{
    VNCViewer1.Password = "";
}

function buttonDisconnect_onclick()
{
        VNCViewer1.Stop();
}

function window_onload()
{
        // load settings
        load();
}

// Save settings to the cookie.
function save()
{
        var date = new Date();

        // cookie valid for 2 months
        date.setMonth(date.getMonth() + 2);

        setCookie("server", textServer.value, date, null, null, null);
        setCookie("sendkeys", checkSendKeys.checked, date, null, null, null);
        setCookie("sendmouse", checkSendMouse.checked.value, date, null, null,
null);
        setCookie("zoom", zoom.value, date, null, null, null);
}

// Load settings from the cookie.
function load()
{
        var v;

        v = getCookie("server");
        if (v) textServer.value = v;

        v = getCookie("sendkeys");
        if (v) checkSendKeys.value = v;

        v = getCookie("sendmouse");
        if (v) checkSendMouse.value = v;

        v = getCookie("zoom");
        if (v)
        {
                zoom.value = v;
        }
}

function setCookie(name, value, expires, path, domain, secure)
```

```
{
 var curCookie = name + "=" + escape(value) +
   ((expires) ? "; expires=" + expires.toGMTString() : "") +
   ((path) ? "; path=" + path : "") +
   ((domain) ? "; domain=" + domain : "") +
   ((secure) ? "; secure" : "");

 document.cookie = curCookie;
}

function getCookie(name)
{
 var dc = document.cookie;
 var prefix = name + "=";
 var begin = dc.indexOf("; " + prefix);

 if (begin == -1)
 {
  begin = dc.indexOf(prefix);
  if (begin != 0) return null;
 }
 else
 {
  begin += 2;
 }

 var end = document.cookie.indexOf(";", begin);

 if (end == -1)
 {
  end = dc.length;
  }

 return unescape(dc.substring(begin + prefix.length, end));
}

function deleteCookie(name, path, domain)
{
 if (getCookie(name))
 {
  document.cookie = name + "=" +
  ((path) ? "; path=" + path : "") +
  ((domain) ? "; domain=" + domain : "") +
  "; expires=Thu, 01-Jan-70 00:00:01 GMT";
 }
}
```

```
function fixDate(date)
{
  var base = new Date(0);
  var skew = base.getTime();

  if (skew && 0)
  {
    date.setTime(date.getTime() - skew);
  }
}



//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1
EVENT=Connected(ServerName)>
<!--
 VNCViewer1_Connected(ServerName)
//-->
</SCRIPT>
<SCRIPT returnvalue="true" LANGUAGE=javascript FOR=VNCViewer1
EVENT="Password">
<!--
        VNCViewer1_Password();
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=VNCViewer1 EVENT=Disconnected>
<!--
 VNCViewer1_Disconnected()
//-->
</SCRIPT>
<p>&nbsp;</p>

</BODY>
</HTML>
```

Program 11 VNCX with NetMeeting

Lines in bold are added by me to provide the Interface with NetMeeting capability.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>NetMeeting</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-1252">
<META content="MSHTML 6.00.2800.1400" name=GENERATOR></HEAD>
<BODY text=#666666 vLink=#ff0080 aLink=#ff00ff link=#8080ff bgColor=#FFFFFF
onload="" >
<p align=center> <object id=NetMeeting classid=CLSID:3E9BAF2D-7A79-11d2-9334-
0000F875AE17 >


  </object>
  <br>
  </BODY>
</HTML>
```

Program 12 NetMeeting embedded in an HTML document

Appendix C Prototype of Awareness Tool

```
import java.awt.*;
 import java.net.*;
 import java.io.*;

 public class AwarenessClient extends Frame {

    //Variables for UI
     Button btnServer;
     TextField txtInput;

    //Initialize Output Streams
    OutputStream rawDataOut = null;
    PrintStream DataOut = null;

    //Initialize Socket
    Socket theClientSocket = null;

    AwarenessClient() {

        //Create the Interface
        super("The Awareness Client");
        setLayout(new FlowLayout());

        add(txtInput);
        btnServer=new Button("Inform Server!");
        add(btnServer);
        pack();
        show();
        super.resize(700, 100);
  try {
            //Instantiate a new Socket
            theClientSocket = new Socket("127.0.0.1", 1055);

            rawDataOut = theClientSocket.getOutputStream();
            DataOut = new PrintStream(rawDataOut);

        } catch( UnknownHostException e) {
            System.out.println("Unable to find the Server. Error " +
e);

        } catch ( IOException e ) {
            System.out.println("An IO error has been raised. Error "
+ e);
        }
    }

    public boolean handleEvent(Event event) {

    if (event.id == Event.ACTION_EVENT && event.target == btnServer)
{
        clickedBtnServer();
        return true;
    } else if (event.id == Event.WINDOW_DESTROY) {
        System.exit(0);
        return true;
    }
    return super.handleEvent(event);
    }

    public static void main(String argv[]) {

        new AwarenessClient();
    }

    public void clickedBtnServer() {

        // Send Data to Server
        try {
            DataOut.print(txtInput.getText());
            //Close the Socket
               theClientSocket.close();
        } catch(Exception e) {}
    }
}
```

program 1: AwarenessClient

```
import java.io.*;
 import java.net.*;
 import java.awt.*;

public class AwarenessServer extends Frame {

    TextArea txtPane;

     AwarenessServer() {

      super("The Awareness Server");

       ServerSocket theSocketServer = null;
       Socket socketReturn = null;
       InputStream rawDataIn = null;

  try {

    theSocketServer = new ServerSocket( 1055 );

    //Listen at the port 1055
        socketReturn = theSocketServer.accept();

    //Connection has been achieved with client

     //InetAddress myself = socketReturn.getInetAddress.getHostAddress();

    //Get data from the Socket
    rawDataIn = socketReturn.getInputStream();
    DataInputStream DataIn = new DataInputStream(rawDataIn);

    //Print Data our in the Statue Window
    String Value = DataIn.readLine();
 txtPane = new TextArea("Handover Status\n\n", 10, 50);
      add("Center", txtPane);
      pack();
      show();

//txtPane.appendText ("the Mobile's Client IP Address : " + myself+ "\n\n");


txtPane.appendText ( " the Mobile Client is : " +
socketReturn.getInetAddress().getHostAddress() + ':'+ socketReturn.getPort()+
"\n\n");


txtPane.appendText("" + socketReturn.getInetAddress().getHostAddress() +  "  will
Disconnect due to handover or a weak link \n\n");
 txtPane.appendText(" The Maximum disconnection time is 40 seconds! \n");
    }
     catch( UnknownHostException e) {
      txtPane.appendText("Unable to find Server. Error" + e);
     }
     catch( IOException e ) {
      txtPane.appendText("IO Error has been raised. Error " + e);
    }
  }

  public static void main(String argv[]) {

      new AwarenessServer();
  }

  public boolean handleEvent(Event event) {

    if(event.id == Event.WINDOW_DESTROY) {

      System.exit(0);
      return true;
    }
    return super.handleEvent(event);
  }
}
```

program 2 Awareness Server

Appendix D Implementation of Freeze-TCP

➢ Difficulties with Linux:

Different problems arose while recompiling and setting the linux kernel 2.4, one of which is setting up the X server, which is important for the collaborative environment supplied by VNC, therefore one of the ways to overcome this problem is to use the Xconfigurator or edit the XFree86 configuration file XF86Config or XF86Config-4 in /etc/X11.

Another problem was setting up Ethernet and network connection, where the kernel did not have the Ethernet card drivers for the connection to run, therefore I had to install module e100 , which is the Ethernet card driver specific to the machine I was working on, because it was not available and whenever I try to set up an Ethernet connection it fails because it could not initialize e100 (which is the Ethernet card driver), downloaded the driver linux* 10/100 Adapter Base Driver [e100-3.5.10.tar.gz] from www.intel.com in /home/dana/

1- tar xfz e100-3.5.10.tar.gz
2- cd e100-3.5.10.tar.gz/src
3- compile the driver: make install
4- install the module: modprobe e100
5- add the following line to /etc/modules.conf:
alias eth100 e100

to test if the system has the driver

```
#service network stop
#insmod e100
If working then
#service network start
Important files for network
/etc/sysconfig/network-scripts/ifcfg-eth0, /etc/sysconfig/network
```

To start a TCP/IP network services on an interface

Activate :/sbin/ifup eth0 or ifconfig eth0 up

Deactivate: /sbin/ifdown eth0 r ifconfig eth down

➢ Modification to the kernel source code

The entire kernel source code of Linux Red Hat 7.3 is located at the directory "/usr/src/linux-2.4.18-3". To implement Freeze-TCP reported in this thesis, I have modified the kernel source codes based on the work made in [**Error! Reference source not found.**], recompiled and made a new kernel for the Linux operating system.

It is wise to build a new kernel that is separated from the old kernel in case the new one fails by doing the following:

1. cp –a /usr/src/linux-2.4.18-3 ~dana/ (where dana is my home path)
2. su rm –f /usr/src/linux
3. su ln –s ~dana/linux-2.4.18-3 /usr/src/linux
4. su chown –R dana:root ~dana/linux-2.4.18-3
5. chmod –R o-rwx ~dana/linux-2.4.18-3

1. Adding a System call:

Then we edit the kernel source code where we need to add Freeze-TCP as a system call[1] by implementing the following steps, I will give a brief description of the process of adding a general new system call before going into details of implementing Freeze-TCP:

1. Define a new system call you insert a new entry in interrput table by editing the file linux/arch/i386/kernel/entry.S.

```
 .long SYMBOL_NAME(sys_myservice)
e.g.        .long SYMBOL_NAME(sys_ftcp)
```

2. You also need to generate a system call "stub" so that an ordinary user program can invoke your system call. You do this by editing the file linux/include/asm/unistd.h
   #define __NR_myservice 191 (with the number interchangeable depending on the last system call number where we add 1)
   e.g. #define __NR_ftcp 239

3. Define (or implement) the system call, have the system call definitions in your own source code files myservice.h and myservice.c, e.g. the files ftcp.h and ftcp.c source code is mentioned below.
4. Modify the Makefile in the directory we placed our .c file e.g. (~linux-2.4.18-2/net/ipv4) so that your code gets compiled and linked in properly. Modify the Makefile line to have a .o of your source code. For example, adding myservice.o:

```
O_OBJS += ... myservice.o
```

In our case also the Config.in file must be edited to add Freeze-TCP as an option the user can choose when configuring the kernel in networking options.

3. Recompile and build a new kernel

After all of the source code is modified, the kernel of Linux is ready to compile to activate Freeze-TCP for the operating system.

First to make sure there are no stale .o files laying around type:

[1] System calls are explicit request to the kernel made via software interrupts. System calls can be taken as interface given to the user mode process to access kernel or device specific data.

1. make mrproper
2. make clean (to remove old object files)

Then configure the kernel:

3. make xconfig (or make config or make menuconfig)

 Select Load Configuration from file (/usr/src/linux-2.4.18-3/configs/kernel-2.4.18-i386.config)

 Select "yes" for TCP: Freeze-TCP support under Networking options

 Save and Exit

4. make dep (to set up all the dependencies correctly)

Then compile:

5. nohup make bzImage & (to build the new kernel)
6. tail –f nohup.out
7. make modules (requires root permission and doing it will delete the currently installed kernel modules located under /lib/moduls/2.4.18-3)
8. make modules_install
9. cp arch/i386/boot/bzImage /boot/bzImage-2.4.18-FTCP
10. mkinitrd -v /boot/initrd-2.4.18-FTCP.img 2.4.18-FTCP

May need the following symbolic links:

11. cd /home/dana/linux-2.4.18-3/
12. cp System.map /boot/System.map-2.4.18-FTCP
13. cp module.info /boot/module.info-2.4.18-FTCP

Then

14. cd /boot
15. ln –fs System.map-2.4.18-FTCP System.map
16. ln –fs module.info-2.4.18-FTCP module.info

To make it work on dual boot system:

17. gedit or vi /boot/grub/menu.lst

title Red Hat Linux (Freeze-TCP)
root (hd0,2)
kernel /bzImage-2.4.18-3-FTCP ro root=/dev/hda5
initrd /initrd-2.4.18-122005.img

4. Running Freeze-TCP

To run Freeze TCP we should create a file to invoke the Freeze TCP system call, we will name the file freeze.c[2] and the source code is in appendix B:

And we compile this program by executing the following command:

cc freeze.c -o freeze –I/home/dana/linux-2.4.18-3/include/net (the directory that contains ftcp.h)

[2] Source code from http://www.csee.umbc.edu/~phatak/sw/ftcp/index.html

5. Implementing Freeze-TCP :
1. /home/dana/linux-2.4.18-3/Makefile
Line 4 EXTRAVERSION = -FTCP (editing this line will ensure that the modules are not overwritten and a new folder 2.4.18-FTCP is created)

Line 98 ifdef CONFIG_FTCP
Line 99 CFLAGS += -D_FTCP_
Line 100 endif

2. /home/dana/linux-2.4.18-3/arch/i386/kernel/entry.S
Line 647 .long SYMBOL_NAME(sys_kill)
Line 648 #ifdef _FTCP_
Line 649 .long SYMBOL_NAME(sys_ftcp)
Line 650 #endif /* _FTCP_ */

3. /home/dana/linux-2.4.18-3/arch/i386/kernel/Makefile
Line 10 ifdef CONFIG_FTCP
Line 11 FTCP = -D_FTCP_
Line 12 endif
Line 13
Line 14 .S.o:
Line 15 $(CC) $(FTCP) $(AFLAGS) -traditional -c $< -o $*.o

4. /home/dana/linux-2.4.18-3/include/asm-i386/unistd.h
We add the system call at the end of the file
Line 247 #ifdef _FTCP_
Line 248 #define __NR_ftcp 239 (add 1 to the number of the last call)
Line 249 #endif /* _FTCP_ */

5. /home/dana/linux-2.4.18-3/include/linux/netdevice.h
Line 430 #ifdef _FTCP_
Line 431 unsigned char ftcp;
Line 432 #endif /* _FTCP_ */

6. /home/dana/linux-2.4.18-3/include/net/ftcp.h (new file)

```
#ifndef _FTCP_H
#define _FTCP_H
#define _FTCP_

#indef __NR_ftcp
#define __NR_ftcp               239
#endif
```

```
#define ftcp(func,fp) syscall(__NR_ftcp, (func), (fp))

#define FTCP_FREEZE (0x01)
#define FTCP_IFOFF (0x02)


typedef struct {
char *ifname;
int val;
} ftcp_t;

enum {
FTCP_FREEZE_CHECK,
FTCP_FREEZE_OFF,
FTCP_FREEZE_ON,
FTCP_IF_CHECK,
FTCP_IF_DISABLE,
FTCP_IF_ENABLE,
FTCP_DUP_ACKS1,
FTCP_DUP_ACKS3
};

/* function prototypes */
void ftcp_freeze_check(ftcp_t *fp);
void ftcp_freeze_on(ftcp_t *fp);
void ftcp_freeze_off(ftcp_t *fp);
void ftcp_if_check(ftcp_t *fp);
void ftcp_if_disable(ftcp_t *fp);
void ftcp_if_enable(ftcp_t *fp);
void ftcp_send_dup_acks(ftcp_t *fp, int num);
#endif /* _FTCP_H */
```

7./home/dana/linux-2.4.18-3/net/core/dev.c

```
Line 110 #ifdef _FTCP_
Line 111 #include <net/ftcp.h>
Line 112 #endif /* _FTCP_ */
Line 741 #ifdef _FTCP_
Line 742 /* Initialize freeze TCP stuff */
Line 743 dev->ftcp = 0;
Line 744 #endif /* _FTCP_ */
Line 1015 #ifdef _FTCP_
Line 1016 /* Discard sk_buff if interface is off */
Line 1017 if (dev->ftcp & FTCP_IFOFF)
Line 1018 {
Line 1019 kfree_skb(skb);
Line 1020 return -ENOMEM;
Line 1021 }
Line 1022
Line 1023 /* Set advertised window size to zero if freeze TCP is on */
Line 1024 if ((skb->sk != NULL) && (skb->sk->protocol == IPPROTO_TCP) &&
Line 1025 (dev->ftcp & FTCP_FREEZE))
Line 1026 skb->h.th->window = 0;
Line 1027 #endif /* _FTCP_ */
Line 1250 #ifdef _FTCP_
Line 1251 /* Discard sk_buff if interface is off */
Line 1252 if (skb->dev->ftcp & FTCP_IFOFF)
Line 1253 {
Line 1254 kfree_skb(skb);
Line 1255 return softnet_data[this_cpu].cng_level;
Line 1256 }
Line 1257 #endif /* _FTCP_ */
```

8./home/dana/linux-2.4.18-3/net/ipv4/Config.in

```
Line 4 bool ' TCP: Freeze TCP support' CONFIG_FTCP
```

9. /home/dana/linux-2.4.18-3/net/ipv4/ftcp.c (new file)

```
#include <net/ftcp.h>
#include <net/tcp.h>
#include <linux/sched.h>
#include <linux/errno.h>
/* ftcp system call */
asmlinkage int sys_ftcp(int func, ftcp_t *fp)
{
if (!capable(CAP_NET_ADMIN))
return -EACCES;
switch (func) {
case FTCP_FREEZE_CHECK:
ftcp_freeze_check(fp);
break;
case FTCP_FREEZE_OFF:
ftcp_freeze_off(fp);
break;
case FTCP_FREEZE_ON:
ftcp_freeze_on(fp);
break;
case FTCP_IF_CHECK:
```

```
ftcp_if_check(fp);
break;
case FTCP_IF_DISABLE:
ftcp_if_disable(fp);
break;
case FTCP_IF_ENABLE:
ftcp_if_enable(fp);
break;
case FTCP_DUP_ACKS1:
ftcp_send_dup_acks(fp, 1);
break;
case FTCP_DUP_ACKS3:
ftcp_send_dup_acks(fp, 3);
break;
default:
return -EINVAL;
}
return 0;
}
/* Check if freeze TCP is on for an interface */
void ftcp_freeze_check(ftcp_t *fp)
{
/* Structure name change from device to net_device */
struct net_device *d;
/*
dev_get_by_name return a structure but
dev_get change to return an integer
*/
if ((d = dev_get_by_name(fp->ifname)) == NULL)
{
fp->val = -ENODEV;
return;
}
fp->val = (d->ftcp & FTCP_FREEZE) ? 1 : 0;
return;
}
/* Turn freeze TCP on for an interface */
void ftcp_freeze_on(ftcp_t *fp)
{
struct net_device *d;
if ((d = dev_get_by_name(fp->ifname)) == NULL)
{
fp->val = -ENODEV;
```

```
return;
}
fp->val = (d->ftcp & FTCP_FREEZE) ? 1 : 0;
d->ftcp |= FTCP_FREEZE;
return;
}
/* Turn freeze TCP off for an interface */
void ftcp_freeze_off(ftcp_t *fp)
{
struct net_device *d;
if ((d = dev_get_by_name(fp->ifname)) == NULL)
{
fp->val = -ENODEV;
return;
}
fp->val = (d->ftcp & FTCP_FREEZE) ? 1 : 0;
d->ftcp &= ~FTCP_FREEZE;
return;
}
/* Check if an interface is disabled */
void ftcp_if_check(ftcp_t *fp)
{
struct net_device *d;
if ((d = dev_get_by_name(fp->ifname)) == NULL)
{
fp->val = -ENODEV;
return;
}
fp->val = (d->ftcp & FTCP_IFOFF) ? 1 : 0;
return;
}
/* Disable an interface */
void ftcp_if_disable(ftcp_t *fp)
{
struct net_device *d;
if ((d = dev_get_by_name(fp->ifname)) == NULL)
{
fp->val = -ENODEV;
return;
}
fp->val = (d->ftcp & FTCP_IFOFF) ? 1 : 0;
d->ftcp |= FTCP_IFOFF;
return;
```

```
}
/* Endable an interface */
void ftcp_if_enable(ftcp_t *fp)
{
struct net_device *d;
if ((d = dev_get_by_name(fp->ifname)) == NULL)
{
fp->val = -ENODEV;
return;
}
fp->val = (d->ftcp & FTCP_IFOFF) ? 1 : 0;
d->ftcp &= ~FTCP_IFOFF;
return;
}
/* Send duplicate ACKs out on all established TCP connections */
void ftcp_send_dup_acks(ftcp_t *fp, int num)
{
/*extern struct sock *tcp_established_hash[TCP_HTABLE_SIZE];*/
int i;
/*
* Find established tcp connections that are not in TIME_WAIT state
* (1st half of hash table)
*/
/* for(i = 0; i < TCP_HTABLE_SIZE/2; i++) */
for (i = 0; i < tcp_ehash_size; i++)
{
/*struct sock *sk = tcp_established_hash[i];
for (sk = tcp_established_hash[i]; sk != NULL; sk = sk->next) */
struct sock *sk;
struct tcp_ehash_bucket *head;
head = &tcp_ehash[i];
for (sk = head->chain; sk != NULL; sk = sk->next)
{
if (!(sk->reuse))
{
int j;
/* FIXME: Check if this connection uses the given device */
/* Send num duplicate ACKs */
for(j = 0; j < num; j++)
tcp_send_ack(sk);
fp->val++;
}
}
}
return;
}
```

10. /home/dana/linux-2.4.18-3/net/ipv4/Makefile

```
Line 21 obj-$(CONFIG_FTCP) += ftcp.o
```

11. /home/dana/linux-2.4.18-3/net/ipv4/tcp_input.c

```
Line 71 #ifdef _FTCP_
Line 72 #include <net/ftcp.h>
Line 73 #endif /* _FTCP_ */
Line 1875 #ifdef _FTCP_
Line 1876 /* Sender side fix for implemenatation of Freeze TCP */
Line 1877 if( nwin == 0 )
Line 1878 tp->snd_wnd = 0;
Line 1879 /* End of sender side fix */
Line 1880 #endif /* _FTCP_ */
Line 2589 #ifdef _FTCP_
Line 2590 if(!(((tp->send_head != NULL) &&
Line 2591 (tp->send_head->dev->ftcp & FTCP_FREEZE)) &&
Line 2592 (((TCP_SKB_CB(skb)->end_seq)-(tp->rcv_nxt)) <= 1)))
Line 2593 #endif /* _FTCP_ */
Line 3032 #ifdef _FTCP_
Line 3033 /* Out receive window == 0 (Zero Window Probe) JM --fix for ZWP */
Line 3034 (tcp_receive_window(tp) == 0) ||
Line 3035 #endif /* _FTCP_ */
Line 3339 #ifdef _FTCP_
Line 3340 /* Fix to zero window probe */
Line 3341 if ((tcp_receive_window(tp) != 0) &&
Line 3342 (!(((tp->send_head != NULL) &&
Line 3343 (tp->send_head->dev->ftcp & FTCP_FREEZE))
Line 3344 && ((TCP_SKB_CB(skb)->end_seq-tp->rcv_nxt) <= 1))))
Line 3345 #endif /* _FTCP_ */
```

12. /home/dana/freeze.c

```
 #include <ftcp.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>

#define IFNAME "eth0"          /* Network interface to use. */
```

```
void error(char *msg)
{
 fprintf(stderr, "ftcp: ERROR");
 if (msg != NULL)
   fprintf(stderr, " %s", msg);
 if (errno != 0)
   fprintf(stderr, " [%s]", strerror(errno));
 fprintf(stderr, ".\n");
 exit(1);
}
int main(int argc, char *argv[])
{
 ftcp_t fp;
 fp.ifname = IFNAME;

 /* Check freeze TCP status for network interface IFNAME */
 if (ftcp(FTCP_FREEZE_CHECK, &fp) == -1 || fp.val == -ENODEV)
   error("FTCP_FREEZE_CHECK failed");

 printf("Freeze TCP is %s for interface " IFNAME ".\n",
       (fp.val == 1) ? "enabled" : "disabled");

 /* Enable freeze TCP for network interface IFNAME */
 printf("Enabling Freeze TCP for interface " IFNAME ".\n");
 if (ftcp(FTCP_FREEZE_ON, &fp) == -1 || fp.val == -ENODEV)
   error("FTCP_FREEZE_ON failed");

 /* Wait a short while so packet(s) with zero window advertisement
  *  get sent.
  */
 usleep(500000);

 /* Drop all packets going through IFNAME */
 printf("Disabling network interface " IFNAME ".\n");
 if (ftcp(FTCP_IF_DISABLE, &fp) == -1 || fp.val == -ENODEV)
   error("FTCP_IF_DISABLE failed");

 /* Wait a while */
 sleep(5);

 /* Disable freeze TCP for network interface IFNAME */
 printf("Disabling Freeze TCP for interface " IFNAME ".\n");
 if (ftcp(FTCP_FREEZE_OFF, &fp) == -1 || fp.val == -ENODEV)
```

```
    error("FTCP_FREEZE_OFF failed");

  /* Allow packets back through IFNAME */
  printf("Enabling network interface " IFNAME ".\n");
  if (ftcp(FTCP_IF_ENABLE, &fp) == -1 || fp.val == -ENODEV)
    error("FTCP_IF_ENABLE failed");

  /* Send duplicate ACKs */
  printf("Sending 3 duplicate ACKs on all established TCP connections.\n");
  if (ftcp(FTCP_DUP_ACKS3, &fp) == -1 || fp.val == -ENODEV)
    error("FTCP_DUP_ACKS3 failed");

  exit(0);
}
```