



VIRTUAL NETWORK COMPUTING: IMPLEMENTATION OF A REMOTE DISPLAY SYSTEM

Neeru Mago

Department of Computer Science and Applications, Panjab University Swami Sarvanand
Giri Regional Centre Hoshiarpur, Punjab

Cite This Article: Neeru Mago, "Virtual Network Computing: Implementation of a Remote Display System", International Journal of Current Research and Modern Education, Volume 3, Issue 1, Page Number 129-134, 2018.

Copy Right: © IJCRME, 2018 (All Rights Reserved). This is an Open Access Article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract:

Virtual network computing (V.N.C) is a process of controlling a computer by sitting kilometers away through internet. Here we can control a server computer, which is situated kilometers away by sitting in front of a viewer computer. An image of the desktop of the server is brought to our computer and making events in viewer computer we can do any work in the server computer. Here internet is used as the communication between the server and the viewers computer. As the operating system is graphical user interface the controlling is made by mouse events. When we brought the servers desktop to viewers computer the screen resolution of the server and viewers must be same (eg: 800*600). So when a mouse event is happening on a particular pixel (say 300,200), that particular pixel send to the server from the viewer and the change is brought to the viewer. So the user will not feel that two computers are working here. He will get feeling that he is using the server itself. The V.N.C technique has got a good future in the world computer communication, since the V.N.C is easy to use, support platform independency and multi-user. It is a remote display system which allows you to view a computing desktop environment not only on the machine where it is running, but from anywhere on the Internet and from a wide variety of machine architectures.

Key Words: VNC, TCP/IP, VNC Viewers, VNC Servers & Client/Server Communication

1. Introduction:

VNC consists of two types of component. A server, which generates a display, and a viewer, which actually draws the display on your screen. To get started with VNC you need to run a server, and then connect to it with a viewer. Get the packages for the platforms you use from the web and install them. The current VNC software requires a TCP/IP connection between the server and the viewer. Most people will be running either a Unix server or a Windows server, though similar principles will apply to other platforms. The technology underlying the VNC system is a simple protocol for remote access to graphical user interfaces. It works at the frame buffer level and therefore applies to all operating systems, windowing systems, and applications indeed to any device with some form of communications link. The protocol will operate over any reliable transport such as TCP/IP. The endpoint with which the user interacts (that is, the display and/or input devices) is called the VNC client or viewer. The endpoint where changes to the frame buffer originate (that is, the windowing system and applications) is known as the VNC server. VNC is truly a "thin client" system. Its design makes very few requirements of the client, and therefore simplifies the task of creating clients to run on a wide range of hardware.

Adaptive Update:

A set of rectangles of pixel data makes a frame buffer update (or simply, update). An update represents a change from one valid frame buffer state to another. In this sense, an update is similar to a frame of video. It differs, however, in that it usually affects only a small area of the frame buffer. Each rectangle may be encoded using a different scheme. The server can therefore choose the encoding most appropriate for the particular screen content being transmitted and the available network bandwidth. The update protocol is demand-driven by the client. That is, an update is only sent by the server in response to an explicit request from the client. All screen changes since the clients last request are coalesced into a single update. This gives the protocol an adaptive quality: the slower the client and the network, the lower the rate of updates. On a fast network, for example, as the user drags a window across the screen it will move smoothly, being drawn at all the intermediate positions. On a slower link for example, over a modem the client will request updates less frequently, and the window will appear at fewer of these positions. This means that the display will reach its final state as quickly as the network bandwidth will allow, thus maximizing the speed of interaction.

Input:

The input side of the VNC protocol is based on a standard workstation model of a keyboard and multi button pointing device. The client sends input events to the server whenever the user presses a key or pointer button, or moves the pointing device. Input events can also be synthesized from other nonstandard I/O devices. On the Video tile, for example, a pen-based handwriting recognition engine generates keyboard events.

Connection Setup and Shutdown

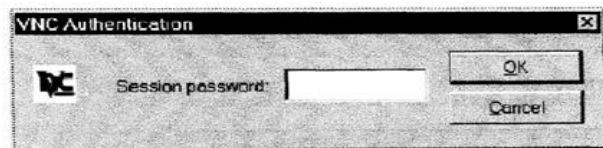
To establish a client-server connection, the server first requests authentication from the client, using a challenge-response VNC server VNC viewer (client) VNC protocol scheme. The client typically requires the user to enter a password at this point. The server and client then exchange messages to negotiate desktop size, pixel format, and encoding schemes. The client requests an update for the entire screen, and the session begins. Because of the stateless nature of the client, either side can close the connection at any time without adverse consequences

VNC Viewers:

In day-to-day use, we prefer the more descriptive term viewer to the rather overloaded word client. Writing a VNC viewer is a simple task, as indeed it should be for any thin client system. It requires only a reliable transport (usually TCP/IP), and a way of displaying pixels (either writing directly to the frame buffer or going through a windowing system). We have written viewers for all the networked display devices available at ORL. These include the Video tile (the original VNC viewer), an X-based viewer (which runs on Solaris, Linux, and Digital Unix workstations), a Win32viewer that runs on Windows NT and 95, and a Java applet that runs on any Java-capable browser (including Sun's Java Station). Members of our lab use these viewers on a daily basis to access their personal computing environments.

Running a Viewer:

You can now go to another machine and connect a viewer to the server. When you run the viewer, you need to specify the name of the server and the number of the desktop. If, for example, you have started a server as display 2 on a machine called „snoopy“. You can start a viewer for it by typing: VNC viewer snoopy:2 With the Windows viewer, you can run it from the command line, but you will more typically run it from the VNC group on the Start Menu .In this case, you will be prompted for the host name and display number. Enter it and click OK, and you will be prompted for your password, after which you



should see the remote display. If you are connecting to a Windows or Mac server, the display number will be 0, unless you have explicitly changed it. If the machine running the server does not have a proper DNS entry, you probably won't be able to use the name and will have to replace snoopy: 2 with something like 192.168.1.2:2. You can get round this on most platforms by creating a hosts" file which maps names onto IP addresses.

VNC Servers:

Writing a VNC server is slightly harder than writing a viewer .Because the protocol is designed to make the client as simple as possible ,it is usually up to the server to perform any necessary translations (for example, the server must provide pixel data in the format the client wants). We have written servers for our two main platforms, X (that is, Unix) and Windows NT/95. The X-based server was the first one we developed. 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 Windows VNC server was a little more difficult to create. Windows has fewer places to insert hooks into the system to monitor display updates, and the model of multi user operation is less clearly defined. Our 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 X-based server, the X viewer, the Win32server, and Win32 viewer can all fit on a single floppy disk. We have also created "thin" servers which produce displays other than desktops, using a simple toolkit. A "VNCCD player," for example, generates a CD player user interface using VNC directly without any reference to a windowing system or frame buffer. Such servers can run on very simple hardware, and can be accessed from any of the standard VNC viewers.

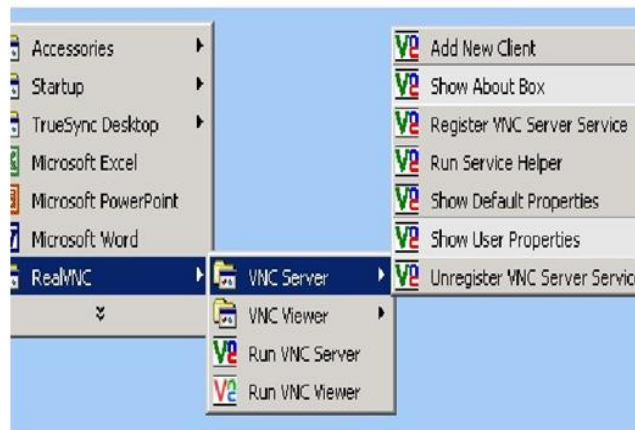
Running a Windows Server:

Win VNC, by running the Setup program included in the distribution. This will create a VNC group in your Start Menu. Install the default registry settings using the option in the VNC group run the Win VNC server. If this is the first time you have used Win VNC on this machine you will be prompted to set a password, which you will need. When you connect to the machine from a remote location. Normally you will want to leave the other options on their default settings (Note that the default display number is 0 on a PC. You



will need to specify this to the viewer when you connect). Click OK and the server should be running. It will install a small icon on the system tray, and by right-clicking on this you can control most aspects of the server.

VNC after Installation:



2. Features of VNC:

No state is stored at the viewer: This means you can leave your desk, go to another machine, whether next door or several hundred miles away, reconnect to your desktop from there and finish the sentence you were typing. Even the cursor will be in the same place. With a PC X server, if your PC crashes or is restarted, all the remote applications will die. With VNC they go on running. Small and simple: The Win32 viewer, for example, is about 150K in size and can be run directly from a floppy. There is no installation needed. Truly platform independent: A desktop running on a Linux machine may be displayed on a PC .Or a Solaris machine. Or any number of other architectures. The simplicity of the protocol makes it easy to port to new platforms. We have a Java viewer, which will run in any Java capable browser. We have a Windows NT server, allowing you to view the desktop of a remote NT machine on any of these platforms using exactly the same viewer. Another people have ported VNC to a wide variety of other platforms. Click the “Contributed” button on the left for details. Sharable: One desktop can be displayed and used by several viewers at once, allowing CSCW-style applications. Free: You can download it, use it, and redistribute it under the terms of the GNU Public License Both binaries and source codes are available from the download page along with a complete copy of this documentation.

3. Advance Features of VNC:

We find VNC to be perfectly acceptable as our normal method of accessing Unix desktops on a daily basis. This is over a 10 M bits Ethernet on reasonably modern machines, using the X or Win32 viewer. Because Windows gives us fewer hints about what it's doing, and because we don't have the source code for Windows in the same way that we do for X, the Win VNC server has to work harder to find out what changed, and so a really fast machine should make a big speed difference. For more information about how the Windows server works. But if you have been disappointed by the speed of the Windows server, don't give up. We are improving it gradually, but it will be a while before it's as fast as on Unix. There are several things that can slow any VNC session down, however, and you may like to consider these if you find it too slow: Unusually busy desktops. The VNC protocol is very efficient a trending areas of a single colour, such as you generally find on window title bars, scrollbars, backgrounds of pages etc. But if, for example, you have pretty 24-bit photographs of your girlfriend as your screen background, or dithered title-bars on your windows, you may pay a price for the aesthetics. A colourful or patterned desktop background will probably slow down VNC more than any other single factor. Access to your VNC desktop generally allows access to your whole environment, so security is obviously important. VNC uses a challenge-response password scheme to make the initial connection the server sends a random series of bytes, which are encrypted using the password typed in, and then returned to the server, which checks them against the right answer. After that the data is unencrypted and could, in theory, be watched by other malicious users, though it's a bit harder to snoop a VNC session.

4. Applications of VNC:

Access important information which you cannot take with you because it is frequently changing or not portable. Check-up emails and download. Start/stop any processing or computation if you forgot to do that in the office. Make simple actions or show “how to” collaborating with people in your office. At ORL, we have used VNC to add mobility to workstation, GUIs, where the concept of at least some form of remote interaction is not new.1.



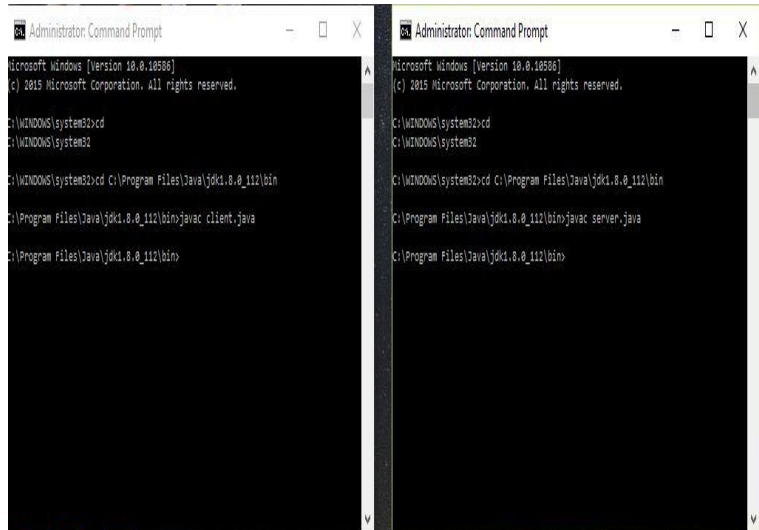
Bring up the controls for their video recorder on a mobile phone as they drive home from work 2. Use a modem to dial a telephone answering machine and reprogram it through a graphical interface 3. Display their car stereo or GPS receiver as part of the dashboard, regardless of the equipment brand installed.



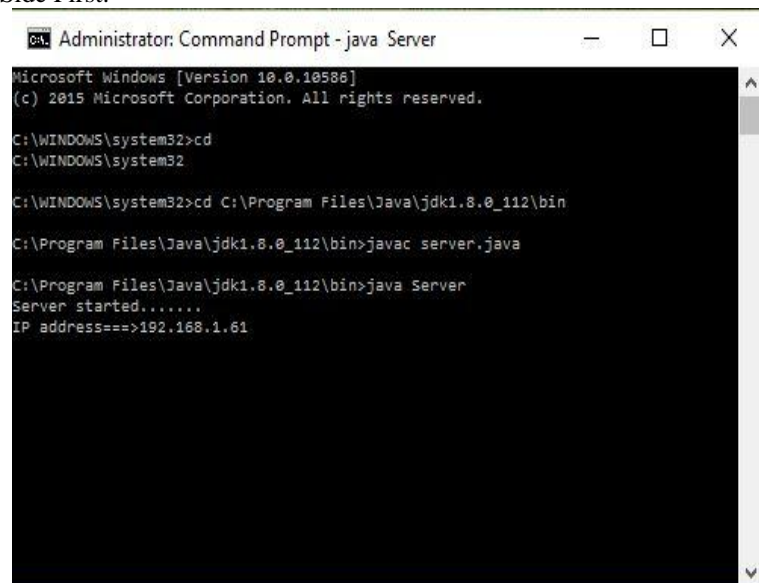
<p>Advantages of VNC</p> <p>Eliminate distance constrain.</p> <p>Automated trouble shooting diagnosing and repairing</p> <p>Saves energy.</p> <p>Virtual files are easy to backup.</p>	<p>Disadvantages of VNC</p> <p>A one way communication for cellular phones</p> <p>Unidirectional flow of control</p> <p>Expensive.</p> <p>RFB 3.X or lower.</p>
--	---

Implementation to Understand Connection Oriented Client/Server Communication:

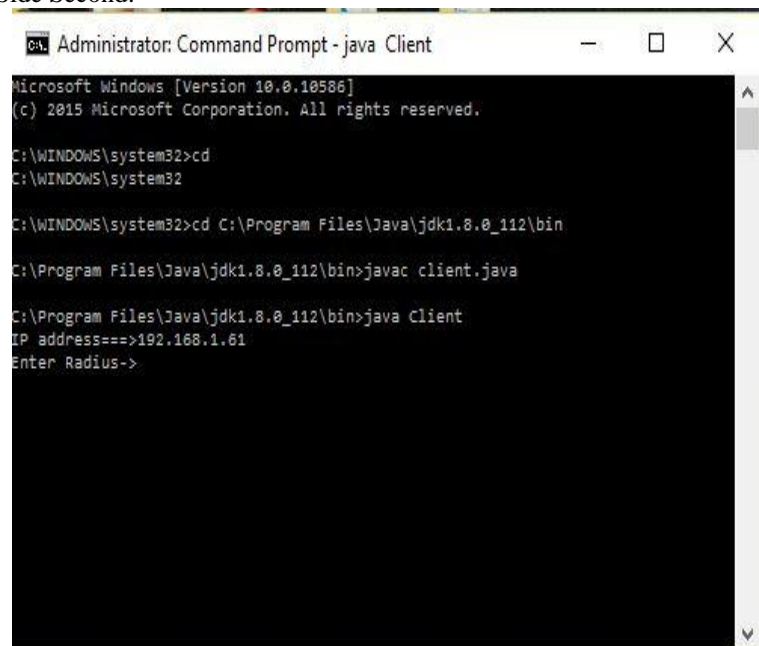
<p>5.1 create a client program</p> <pre>import java.io.*; import java.net.*; import java.util.Scanner; class Client{ double radius,result; Client() { try{ Socket clientSoc=new Socket("127.0.0.1",4000); InetAddress addr=InetAddress.getLocalHost(); System.out.println("IP address====>" +addr.getHostAddress()); Scanner input=new Scanner(System.in); System.out.println("Enter Radius-> "); radius=input.nextDouble(); DataInputStream fromServ=new DataInputStream(clientSoc.getInputStream()); DataOutputStream toServ=new DataOutputStream(clientSoc.getOutputStream()); toServ.writeDouble(radius); System.out.println("Radius sent to server :"+radius); toServ.flush(); result=fromServ.readDouble(); System.out.println("Area of circle (radius="+radius+")==>" +result); fromServ.close(); toServ.close(); clientSoc.close(); } catch(Exception e){ e.printStackTrace(); } } public static void main(String[] ar){ new Client();} }</pre>	<p>5.2 create a server program</p> <pre>import java.net.*; import java.io.*; class Server{ double area; Server() { try { ServerSocket serverSock=new ServerSocket(4000); System.out.println("Server started....."); InetAddress addr=InetAddress.getLocalHost(); System.out.println("IP address====>" +addr.getHostAddress()); while(true){ Socket soc=serverSock.accept(); System.out.println("Local port number of server-- >" +soc.getLocalPort()); DataInputStream din=new DataInputStream(soc.getInputStream()); DataOutputStream dos=new DataOutputStream(soc.getOutputStream()); double radius=din.readDouble(); System.out.println("Radius Received from client is : "+radius); area=Math.PI*radius*radius; System.out.println("Area sent to client is : "+area); dos.writeDouble(area); din.close(); dos.close(); soc.close(); } } catch(Exception e) { e.printStackTrace(); } } public static void main(String[] ar) { new Server(); } }</pre>
--	--



5.3 Compile Server Side First:



5.4 Compile Client Side Second:



5.5 Run the Client Side and See the Result on Server Side Automatically:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd
C:\WINDOWS\system32>
C:\WINDOWS\system32>cd C:\Program Files\Java\jdk1.8.0_112\bin
C:\Program Files\Java\jdk1.8.0_112\bin>javac client.java
C:\Program Files\Java\jdk1.8.0_112\bin>java Client
IP address==>192.168.1.61
Enter Radius->
4
Radius sent to server :4.0
Area of circle (radius=4.0) ==>50.26548245743669

C:\Program Files\Java\jdk1.8.0_112\bin>

Administrator: Command Prompt - java Server
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd
C:\WINDOWS\system32>
C:\WINDOWS\system32>cd C:\Program Files\Java\jdk1.8.0_112\bin
C:\Program Files\Java\jdk1.8.0_112\bin>javac server.java
C:\Program Files\Java\jdk1.8.0_112\bin>java Server
Server started.....
IP address==>192.168.1.61
Local port number of server->4000
Radius Received from client is : 4.0
Area sent to client is : 50.26548245743669
```

6. Future Scope:

We are now building VNC software for a variety of desktop platforms, but it would not be difficult to make remote access practical for a wider range of devices. We can envisage cheap hardware that might, for example, drive a 7-segment LCD and also emit a VNC equivalent over a USB or RS232 link. The VNC commands to draw and erase each Segment could be stored as a sequence of bytes in a small amount of ROM and sent over a communications link when the segment is lit or switched off. Hardware such as this, if made in quantity, could be very cheap and could allow for mobility of much more than just a conventional “desktop.” If built into television sets, VNC viewers could allow them to act as displays for a very wide range of devices including, of course, the PC at the office.

7. Conclusion:

VNC is a very impressive product, especially considering that it's free. Its main advantage over its commercial competitors is that it is open source. So anyone with programming skills can contribute towards it and so make it an even better more flexible product. Even in its current form, its remote control applications are almost limitless, and it well no doubt finds many users and supporters in a typical support environment. VNC means really here, virtually there.

8. References:

1. Tomar et.al "Virtual Network Computing- a Prodigious Technology for Remote Desktop Sharing", Int. Journal of Engineering Research and Applications www.ijera.com ISSN: 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.59-63.
2. Murarka et.al “Virtual Network Computing Based Droid desktop”, IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 5 (Nov. - Dec. 2012), PP 16-20.
3. Patel et.al " Mobile Virtual Network Computing System", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 3, March 2014.
4. <http://www.realvnc.com/docs/rfbproto.pdf>
5. Internet Engineering Task Force (IETF) T. Richardson, J. Levine Category: Informational Real VNC Ltd. ISSN: 2070-1721 March 2011
6. <https://www.slideshare.net/TheGreatLeo/virtual-network-computing>
7. https://en.wikipedia.org/wiki/Virtual_Network_Computing
8. <http://www.remoteaccess.org/what-is-a-vnc/>
9. <https://www.techopedia.com/>
10. <https://www.slideshare.net/rojalina030/virtual-network-computing>
11. Sanaullah Baig et.al " Virtual Network Computing Based Remote Desktop Access", International Journal of Computer Science and Telecommunications, Volume 3, Issue 5, May 2012.