ISSN 2047-3338

Virtual Network Computing Based Remote Desktop Access

Md. Sanaullah Baig[1], Rajasekar M.[2] and Balaji P.[3]

[1,2,3]Department of Computer Science and Engineering, Gojan School of Business and Technology, Chennai-600052, India

[1]sanubaig@hotmail.com, [2]rerajasekar@gmail.com, [3]balaji291289@gmail.com

***Abstract*– Cloud computing means using multiple server computers via a digital network as though they were one computer. Often, the services available is considered part of cloud computing. Cloud computing broadens the range of applications offered to mobile end-users with demanding applications in terms of graphical hardware, such as 3D virtual environments, or storage capacity, such as 3D medical imaging applications. As the cloud infrastructure is shared among multiple users, these hardware resources can be provided in a cost-effective way. Mobile cloud computing can give mobile device users a number of advantages. Company users are able to share resources and applications without a high level of capital expenditure on hardware and software resources. Mobile cloud computing provides a solution to meet the increasing functionality demands of end-users, as all application logic is executed on distant servers and only user interface functionalities reside on the mobile device. The mobile device acts as a remote display, capturing user input and rendering the display updates received from the distant server. Essentially, the principle of mobile cloud computing physically separates the user interface from the application logic. Varying wireless channel conditions, short battery lifetime and interaction latency introduce major challenges for the remote display of cloud applications on mobile devices. A number of adequate solutions that have recently been proposed to tackle the main issues associated with the remote display of cloud services on mobile devices.**

***Index Terms*– VNCserver, SVNCserver, VNC Viewer, Remote Frame Buffer Protocol (RFB) and Compact Remote Frame Buffer Protocol (CRFB)**

I. INTRODUCTION

CELLULAR phones have shown a dramatic improvement in their functionality to a point where it is now possible to have cellular phones execute Java programs. As a result, cellular users throughout the world are now able to read and write e-mail, browse Web pages, and play Java games using their cellular phones. This trend has prompted us to propose the use of a cellular phone as a device for remotely controlling computers. For example, if a cellular user is able to remotely access computers (such as workstations in offices and personal computers (PCs) in homes) or other networked digital appliances, it would provide the user with the following capabilities:

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

This project presents a virtual network computing (VNC) based architecture for accessing the desktop of remote systems MS-Windows from a cellular phone. It is assumed that the remote computer system is running a VNC server and that it is attached to a network. The cellular user can see and manipulate the desktop on the cellular phone.

The most sophisticated form of remote access enables users on one computer to see and interact with the actual desktop user interface of another computer. Setting up remote desktop support involves configuring software on both the host (local computer controlling the connection) and target (remote computer being accessed). When connected, this software opens a window on the host system containing a view of the target's desktop.

Current versions of Microsoft Windows include Remote Desktop Connection software. However, this software package only supports target computers running Professional, Enterprise or Ultimate versions of the O/S, making it unsuitable for use with many home networks. For Mac OS X computers, the Apple Remote Desktop software package is also designed for business networks and sold separately. For Linux, various remote desktop software exists.

Many remote desktop solutions are based on Virtual Network Computing (VNC) technology. Software packages based on VNC work across multiple operating systems. The speed of VNC and any other remote desktop software can vary, sometimes performing effectively the same as the local computer but other times exhibiting sluggish responsiveness due to network latency.

II. EXISTING SYSTEM

In the existing system, we use Bluetooth connection to access the system contents in a wireless device. Desktop applications are redesigned to operate on mobile hardware platforms in a shorter and portable mode, thereby often losing functionality. To executing typical office applications users connect over a wired local area network to the central server.

Mobile cloud computing can give mobile device users a number of advantages. Company users are able to share resources and applications without a high level of capital expenditure on hardware and software resources. Mobile cloud computing provides a solution to meet the increasing functionality demands of end-users, as all application logic is executed on distant servers and only user interface functionalities reside on the mobile device. The mobile device acts as a remote display, capturing user input and rendering the display updates received from the distant server.

A. Limitation of Existing System

- In the existing system we use Bluetooth to access the system contents in a mobile
- In this, only particular application can to access
- The files of the system can be accessed only within short distances

III. PROPOSED SYSTEM

The principle of mobile cloud computing physically separates the user interface from the application logic. Virtual network computing (VNC) is a desktop sharing system which uses the RFB (Remote Frame Buffer) protocol to remotely control another computer. It transmits the user events from one computer to another relaying the screen updates back in the other direction, over a network using Buffered IO Stream.

In a mobile cloud computing environment, the remote display protocol deliver complex multimedia graphics over wireless links and render these graphics on a resource constrained mobile device. Offloading applications to the cloud is a straight forward way to save on energy consumption because the amount of local processing is reduced. Efficient compression techniques to reduce the amount of exchanged data are done using compression techniques and versatile graphics encoding, downstream data peak reduction and Optimization of upstream packetization overhead (Fig. 1).

IV. PROBLEM FORMULATION

Conventional desktop applications need to be redesigned to operate on mobile hardware platforms, thereby often losing functionality. More demanding applications typically require specific hardware resources that are very unlikely to be available on mobile devices.

The web hosts increasingly powerful computing resources and has evolved to a ever-present computer, offering applications ranging from simple word processors, over all-encompassing enterprise resource planning suites to 3D games, to make availability of such application is not easy.

V. DESIGN OVERVIEW

A. System Architecture

VNC Viewer sends a request for authentication then VNC Server verify and validate the request if the user is valid authenticated then it will give access to view the server. The RFB Protocol is used to transmit the frames from server to SVNC Proxy. CRBF Protocol is used to transmit the frame from proxy server to viewer (Fig. 2).

B. Modules

1) RFB Protocol

RFB (remote frame buffer) is a simple protocol for remote access to graphical user interfaces. Because it works at the frame buffer level it is applicable to all windowing systems and applications of Windows. RFB is the protocol used in VNC (Virtual Network Computing). The remote endpoint where the user sits (i.e., the display plus keyboard and/or pointer) is called the RFB client or viewer. The endpoint where changes to the frame buffer originate (i.e., the windowing system and applications) is known as the RFB server (Fig. 3).

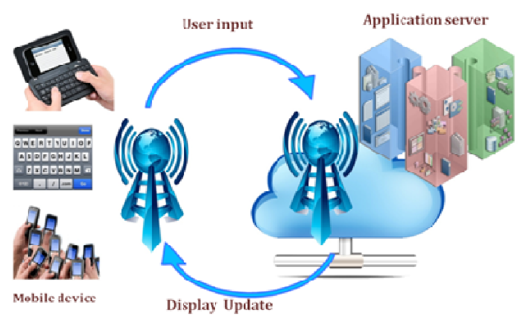


Fig. 1: System Application Overview

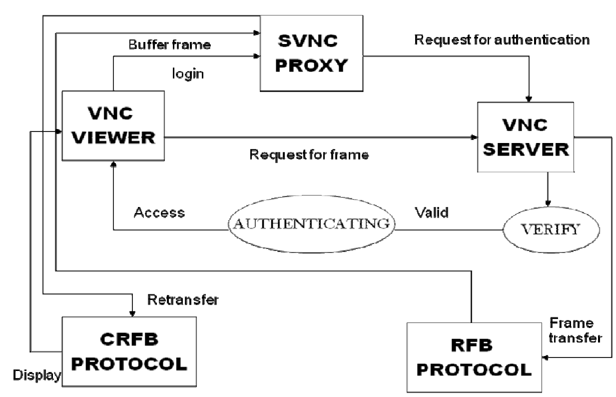


Fig. 2: System Architecture

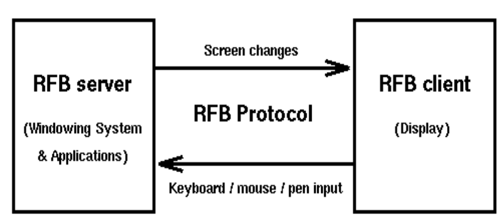


Fig. 3: RFB Protocol Overview

RFB is truly a “thin client” protocol. The emphasis in the design of the RFB protocol is to make very few requirements of the client. In this way, clients can run on the widest range of hardware, and the task of implementing a client is made as simple as possible. The protocol also makes the client stateless. If a client disconnects from a given server and subsequently reconnects to that same server, the state of the user interface is pre- served. Furthermore, a different client endpoint can be used to connect to the same RFB server. At the new endpoint, the user will see exactly the same graphical user interface as at the original endpoint. In effect, the interface to the user's applications becomes completely mobile. Wherever suitable network connectivity exists, the user can access their own personal applications, and the state of these applications is preserved between accesses from different locations. This provides the user with a familiar, uniform view of the computing infrastructure wherever they go.

C. Application Streaming

This platform allows applications to be deployed in real-time to any client from a virtual application server. It removes the need for local installation of the applications. Instead, only the Soft Grid runtime needs to be installed on the client machines. All application data is permanently stored on the virtual application server. Whichever software is needed is streamed from the application server on demand and run locally.

The four files created and installed on the Soft Grid Application Server are accessed by the desktop. The result is the creation of a virtual application environment on the user’s machine with the bare minimum of application components streamed into it. The result is a self-contained application runtime space that virtualizes the following components:

- Registry – registry changes unique to the application are not made to the main OS on the desktop. Rather, they are virtualized within the isolated application runtime space.
- File system – calls from the application for local disk access can be redirected to access DLLs and other components from a virtual file system.
- COM/IPC
- INI files
- Process environment
- Fonts

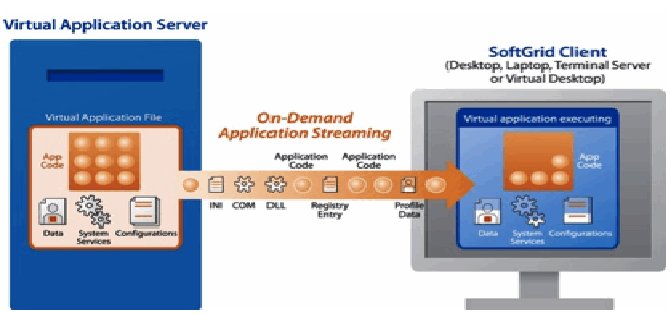


Fig. 4: On-Demand Application Streaming

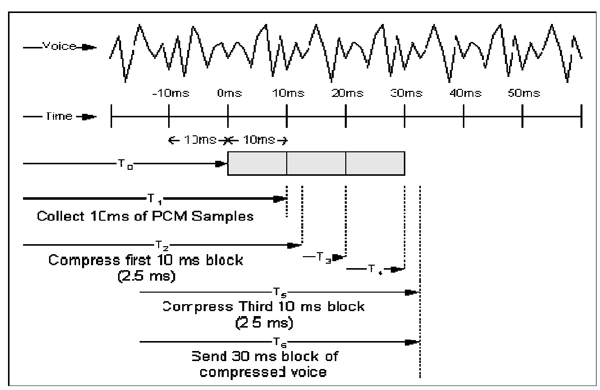
D. Downstream Data Peak Reduction

Interactive applications only update their display unless instructed by the user. Usually, these display updates involve a large amount of data that needs to be sent to the client in a short interval to swiftly update the display Moving or capable of moving with great speed. Their analysis of remote display protocol traffic traces reveals a lot of redundancy, caused by the repainting of graphical objects after recurring user actions.

They propose a hybrid cache-compression scheme whereby the cached data is used as history to better compress recurrent screen updates. The cache contains various drawing orders and bitmaps. Using Microsoft’s Remote Display Protocol (RDP) and dependent on the size of the cache.

E. Optimization of Upstream Packetization Overhead

The some network send in desecrate chunks call packets User events are the principal source of remote display traffic in the upstream direction from client to server. Individually, each user event embodies only a small amount of information: a key or button id, one bit to discriminate between the press and release action and possibly the current pointer coordinates. Nevertheless, user events induce important upstream traffic because they are often generated shortly after each other. Entering a single character results in two user events to indicate the press and release action, a large packetization overhead is observed owing to the headers added at the TCP, IP and (wireless) link layer. The upstream packetization overhead of three commonly used remote display protocols. The maximum buffering period is a consideration of remote display bandwidth reduction against interaction latency the highest bandwidth reductions are achieved for interactive applications with frequent user events and lower roundtrip times screen resolutions, as well as frames per second displayed (Fig. 5).



Fig, 5: The upstream packetization overhead

F. Alternative Methods for Remote Desktop Access

1) Mobile devices such as personal digital assistants, tablet PCs, and cellular phones have greatly enhanced user capability to connect to remote resources. Although a large set of applications is now available bridging the gap between desktop and mobile devices, visualization of complex 3D models is still a task hard to accomplish without specialized

hardware. This paper proposes a system where a cluster of PCs, equipped with accelerated graphics cards managed by the Chromium software, is able to

2) The proposed framework allows mobile devices such as smart phones, personal digital assistants (PDAs), and tablet PCs to visualize objects consisting of millions of textured polygons and voxels at a frame rate of 30 fps or more depending on hardware resources at the server side and on multimedia capabilities at the client side. The server is able to concurrently manage multiple clients computing a video stream for each one; resolution and quality of each stream is tailored according to screen resolution and bandwidth of the client. The paper investigates in depth issues related to latency time, bit rate and quality of the generated stream,

3) Proposed optimization techniques address the major challenges that varying wireless channel conditions, short battery lifetime, and interaction latency pose for the remote display of cloud applications on mobile devices.

4) The Web hosts increasingly powerful computing resources at affordable prices. These resources are not limited to power-hungry hardware but also include diverse software applications that in some cases would overwhelm the connecting client device if they were hosted on the client itself. As Figure 1 shows, such applications range from simple word processors to all-encompassing enterprise resource planning (ERP) suites.

VI. EXPERIMENTAL RESULTS

Before we used Bluetooth technology for mobiles to communicate with desktop system, we have limitations using this method. As we can communicate desktop system only in limited distances. Only limited applications can be performed. Conventional desktop applications are redesigned as a portable view.

So cannot access the full functionality of the desktop system. So we use internet connection and some the protocols are used for effective communication. This process increases the reliability, and able to access the full functionality of the desktop system of anywhere in the world i.e., we can able to access the remote desktop system.

VII. CONCLUSION

By physically separating the user interface from the application logic, the principle of mobile cloud computing allows to access even the most demanding applications in the cloud from intrinsically resource-constrained mobile devices. In this article, we have surveyed contemporary remote display optimization techniques specifically tailored to the short mobile device battery lifetime, the varying and limited bandwidth availability on wireless links and the interaction latency. Although each of these solutions adequately addresses specific challenges of mobile cloud computing, an overall approach is currently lacking. The context of mobile cloud computing is highly dynamic, owing to the user mobility, the wide diversity of applications, and the varying wireless channel status. Future research should therefore be devoted to the design of an overall framework, integrating all the presented solutions, and activating the most appropriate solutions dependent on the current device, network and cloud server status.

REFERENCES

[1] V. S. Pendyala and S. S. Y. Shim, “The Web as the Ubiquitous Computer,” COMPUTER, vol. 42, no. 9, pp. 90–92, SEP 2009.

[2] F. Lamberti and A. Sanna, “A streaming-based solution for remote visualization of 3D graphics on mobile devices,” IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, vol. 13, no. 2, pp. 247–260, MAR-APR 2007.

[3] H. Kawashima, K. Koshiba, K. Tuchimochi, K. Futamura, M. Enomoto, and M. Watanabe, “Virtual PC-type thin client system,” NEC TECHNI-CAL JOURNAL, vol. 2, no. 3, pp. 42–47, SEP 2007.

[4] P. Simoens, F. A. Ali, B. Vankeirsbilck, L. Deboosere, F. De Turck, B. Dhoedt, P. Demeester, and R. Torrea-Duran, “Cross-Layer Optimization of Radio Sleep Intervals to Increase Thin Client Energy Efficiency,” IEEE COMMUNICATIONS LETTERS, vol. 14, no. 12, pp. 1095–1097, DEC 2010.

[5] K.-J. Tan, J.-W. Gong, B.-T. Wu, D.-C. Chang, H.-Y. Li, Y.-M. Hsiao, Y.-C. Chen, S.-W. Lo, Y.-S. Chu, and J.-I. Guo, “A remote thin client system for real time multimedia streaming over VNC,” in 2010 IEEE International Conference on Multimedia and Expo (ICME), 2010, pp. 992–7.

[6] M. Mitrea, P. Simoens, B. Joveski, J. Marshall, A. Taguengayte, F. Pre-teux, and B. Dhoedt, “BiFS-based approaches to remote display for mobile thin clients,” in Proceedings of the SPIE - The International Society for Optical Engineering, vol. 7444, 2009, p. 74440F (8pp.).

[7] G. Paravati, C. Celozzi, A. Sanna, and F. Lamberti, “A Feedback-Based Control Technique for Interactive Live Streaming Systems to Mobile Devices,” IEEE Transactions On Consumer Electronics, vol. 56, no. 1, pp. 190–197, FEB 2010.

[8] A. Boukerche, R. W. N. Pazzi, and J. Feng, “An end-to-end virtual environment streaming technique for thin mobile devices over heterogeneous networks,” Computer Communications, vol. 31, no. 11, pp. 2716–2725, JUL 15 2008.