International Journal of Computational Engineering Research (IJCER) ISSN: 2250-3005
National Conference on Architecture, Software system and Green computing

Solution for Accessing a Computer Using VNC in Mobile Cloud Computing

I.Golda Selia[1] R.Iswarya[2]
[1]Asst Professor/Dept of I.T [2]M.Tech Information Technology
Dr.M.G.R Educational and Research Institute University
Chennai-95 India

Abstract-
This paper discusses about a Virtual Network Computing (VNC) based architecture for accessing a desktop of remote computer from a cellular phone. Mobile cloud computing provides a solution to meet the increasing functionality demands of end users, as all application logic is executed in 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. Varying wireless channel conditions, short battery lifetime and interaction latency introduces major challenges for the remote display of cloud application on mobile devices. In this paper we discuss a number of adequate solution that have recently been proposed to tackle the main issues associated with the remote display of cloud services on mobile devices.

Keywords – VNC, WNIC, ICA, Packetization, Google Apps RFB.

I. Introduction

Cellular Devices have become an essential part of our daily life. Their portability is well appreciated by end-users and Smartphone's sales will soon surpass desktop sales. As these popularity grows, end-user demands to run heavier applications are equally increasing. Although advances in miniaturization continue, the desire to preserve the advantages of weight, size and device autonomy will always impose intrinsic limits on processing power, storage capacity, battery lifetime and display size. Conventional desktop applications need to be redesigned to operate on mobile hardware platforms, thereby often losing functionality; whereas more demanding applications typically require specific hardware resources that are very unlikely to be available on mobile devices. At the same time, the web hosts increasingly powerful computing resources and has evolved to a ubiquitous computer, offering applications ranging from simple word processors, over all-encompassing enterprise resource planning suites to 3D games [1], [2]. Both Microsoft and Google have developed complete online office suites, called Office Live and Google Apps respectively that may evolve to all round alternatives for the mobile office suites. Beyond the conventional office applications, cloud computing broadens the range of applications offered to mobile end-users with demanding applications in terms of

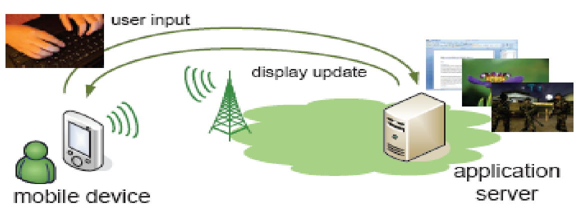

Fig.1 The viewer component on the client forwards the captured user input to the server. In turn, the server side component intercepts, encodes and transmits application output.

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.

Essentially, the principle of mobile cloud computing physically separates the user interface from the application logic. Only a viewer component is executed on the mobile device, operating as a remote display for the applications running on distant servers in the cloud. Any remote display framework is composed of three components: a server side component that intercepts encodes and transmits the application graphics to the client, a viewer component on the client and a remote display protocol that transfers display updates and user events between both endpoints.

II. Mobile Cloud Computing

Mobile cloud computing is a combination between **mobile network** and **cloud computing**, thereby providing optimal services for mobile users. In mobile cloud computing, mobile devices do not need a powerful configuration (e.g., CPU speed and memory capacity) since all the data and complicated computing modules can be processed in the clouds. Mobile computing services have simplified our lives.

III. Vnc Architecture

The Vnc-Based Architecture

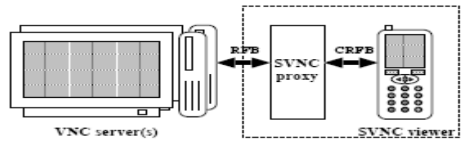

Figure 3. VNC-based architecture

Fig. 2 VNC is an implementation of a remote display system based on a Remote Frame Buffer (RFB) protocol.

STRUCTURE OF VNC

Time considering portability and generality, we propose a VNC based architecture. It consists of VNC servers running on one or more remote computers, a Smart VNC (SVNC) proxy, and a SVNC viewer on a cellular phone. A VNC server sends a remote desktop display as bitmap images in RFB protocol. A SVNC proxy converts (crops, shrinks and resample) 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. The transfer is performed in our own Compact RFB (CRFB), simplified RFB protocol. Then, the SVNC viewer displays the transferred images. Key events received by the SVNC viewer are transmitted to a SVNC proxy that coverts them and sends them to the server. When the user first tries to connect to a remote computer, he must specify his user name and password for authentication as well as the host name of the computer that is running a VNC server. If authentication succeeds, the SVNC proxy establishes a session with the VNC server and the SVNC viewer starts user services. To suppress network traffic, encoding is changed depending on contexts. Usually, colored display images are transferred from the SVNC proxy to the SVNC viewer. However while the user is manipulating the remote desktop, such as scrolling and moving the pointing device, the display images are gray-scaled to reduce the number of bytes required to encode the image.

APPLICATIONS OF VNC

VNC has a wide range of applications including system administration, IT support and helpdesks. It can also be used to support the mobile user, both for hot desking within the enterprise and also to provide remote access. The system allows several connections to the same desktop, providing an invaluable tool for collaborative or shared working in the workplace or classroom. VNC Enterprise Edition is an enhanced version of the industry-standard VNC, developed for use in corporate environments and across the Internet. Designed and built from the ground up by the original inventors of VNC, Enterprise Edition provides robust and easily-administered security with a minimum of fuss.

BENEFITS FROM THE ARCHITECTURE

The VNC protocol is an image-based protocol in which updates to a screen by applications are captured. Therefore, we can manipulate the applications running on the remote system by browsing the same image that we would be browsing if we were sitting at the remote computer. We can utilize the general availability of VNC servers. VNC is becoming widely available as an infrastructure for controlling remote computers and for linking home appliances with PCs due to the portability of the RFB protocol.

Using standard thin client solutions, such as Microsoft Remote Desktop Protocol (RDP), Citrix Independent Computing Architecture (ICA) and Virtual Network Computing (VNC), in a mobile cloud computing context is not straightforward. we present and discuss a number of solutions that have recently been proposed to adequately address these challenges. An overview of the covered solutions is presented in Table1.

TABLE1

Overview of Recent Solutions to the Challenges of Mobile Cloud Computing Surveyed In This Paper.

Challenge	Solutions
Battery Lifetime	Cross-Layer Identification Of WNIC Sleep Opportunities
Wireless Bandwidth	Motion-Based Differential Encoding Individual Object Encoding Real-Time Adaptation Of Encoding Parameters
Interaction Latency	Scene Object caching Buffering Of Key images For Virtual Environments Proximate Hosting Infrastructure Computing Display Updates in Advance

IV. Overcoming the Limited Mobile Device Battery Lifetime

The operational time of mobile devices is often limited when extensively used. These battery capacity shortcomings result in short recharge cycles and refrain users from relying completely on their mobile device. Over the last decade, the advances in nominal battery capacity have been modest. Pentikousis et al. observe that the technological improvements are currently stagnating, because of the lack of a major battery technology breakthrough, comparable to the advent of rechargeable Li-ion batteries. Consequently, extending device autonomy should primarily be realized by making the device itself more energy efficient.

The amount of local processing is reduced. Local processing is however traded off with network bandwidth consumption, and the bidirectional communication with the application server incurs additional drains from the battery to the wireless network interface card (WNIC). From an energy perspective, offloading applications from mobile devices is mainly interesting when large amounts of computation are needed in combination with relatively small amounts of network communication. Demanding applications exchange a significant amount of data between client and server because they exhibit a high degree of interactivity and detailed graphics. According to Kumar's model, this is not beneficial from an energy perspective. On the other hand, as we have described above, these applications have important hardware requirements that are only available in the cloud. To optimize the energy balance, it is important to study the WNIC energy consumption and develop energy optimizing strategies. The WNIC energy consumption is the product of the number of bytes exchanged over the wireless interface, and the energy cost per byte. Efficient compression techniques to reduce the amount of exchanged data are covered in section III of this article. The average energy cost per byte is determined by the distribution of the time over the four possible WNIC states: send, receive, idle and sleep mode. Because in each state a specific set of components is activated, the WNIC power consumption largely differs between the different states.

Figure 3 visualizes our measurements on the average WNIC time distribution in typical remote display scenarios, while Table II contains figures on the typical power consumption in each WNIC state.

In between two display updates, the WNIC is instructed to enter the sleep mode. These sleep modes are only interrupted at regular intervals to transmit user events. Through this cross-layer optimization, WNIC energy consumption reductions up to 52 % have been obtained.

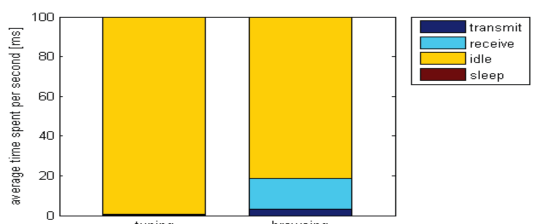

Fig.3. Time distribution of the WNIC mode resulting from remote display protocol traffic in two different scenarios. The WNIC is mainly in idle mode and almost never enters sleep mode.

TABLE-II

Power consumption of the CISCO AIRNET WNIC. *An Energy-Aware Transmission Mechanism For Wifi-Based Mobile Devices Handling Upload Tcp Traffic In InT.*

Power Mode	Measured Power (mW)
Send	1090-155o
Receive	1060-1380
Idle	1150
Sleep	300

V. Advanced Display Compression In A Bandwidth Limited Wireless Environment

Compared with fixed access networks, bandwidth availability on modern broadband mobile and wireless technologies is limited, variable and expensive. Typically, UMTS user receive up to 384 kbps, while Balachandran et al.report practical throughputs of 347 kbps for LTE and up to 6.1 Mbps for WiMAX. Moreover, the actual throughput will vary due to user mobility and interference and fading effects. Besides technological limitations, economical considerations drive the demand for highly efficient remote display compression technologies. More and more, users are confronted with volume based subscription plans and hence will not tolerate any redundant byte to be sent on the network. For example, AT&T, a leading USA service provider, has adopted volume based pricing models in 2010. It is even expected that this economical dimension might impede the development of new cloud applications.

A. Versatile graphics encoding

The choice of codec to compress the intercepted application graphics at the server is a trade off between visual quality, compression efficiency and decoding complexity. Conventional remote display architectures, such as Citrix ICA, AT&T VNC and Microsoft RDP virtualize the graphical library at the server and forward intercepted drawing primitives to the client, such as instructions to draw a rectangle, to display a bitmap or to put some text on the screen. This approach is optimal for applications that only update small regions of the display, or that have a slow refresh rate with respect to the network roundtrip time, such as typical office applications. Bandwidth requirements to remotely display this type of graphics do not exceed 200 kbps, and can be perfectly served over wireless links. On the other hand, a lot of drawing primitives would be required to encode the graphics of multimedia applications, because these update large parts of their screen at high refresh rates and their graphics often contain fine-grained and complex color patterns. This kind of graphics can be more efficiently encoded by means of a video codec, such as H.264 or MPEG- 4 video. Using video codecs for remote display purposes is referred to as interactive live streaming, because the graphics are mainly the result of user interaction, in contrast to regular video streaming with only limited user interaction, e.g. to start and stop the video. Interactive live streaming has been applied successfully in the context of remote 3D virtual environments [2] and gaming. The mobile device regularly reports on the amount of data that is encoded per unit of time, a metric reflecting both the device hardware capabilities, as well as of the amount of data that is received by the device. By adjusting the resolution and image quality accordingly, the controller aims to maintain a target frame rate to ensure a smooth visualization experience

B. 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. The delivery of these data burst requires an instantaneous bandwidth that is much higher than the average bandwidth requirement. Furthermore, this burst traffic pattern is unfavourable in wireless network environments, as it might induce additional collisions on the wireless channel. Sun et al.have studied this problem. 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, they are able to reduce the number of data spikes by 27-42 %, which results in global network traffic reductions of 10-21 %.

C. Optimization of upstream Packetization overhead

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, whereas moving the mouse results in a sequence of pointer position updates. Usually, user events are transmitted as they occur to minimize interaction latency. Because data packets sent upstream often contain a single user event, a large packetization overhead is observed owing to the headers added at the TCP, IP and (wireless) link layer. Table III quantifies the upstream packetization overhead of three commonly used remote display protocols.

The highest bandwidth reductions are achieved for interactive applications with frequent user events and lower roundtrip times. For a text editing scenario and network roundtrip times below 50 ms,Simoens etal.Achieve bandwidth reductions up to 78 %.

TABLE III

Packetization overhead of TCP/IP headers when sending a Single keystroke to the server. The total overhead is further Increased by optional headers and the wireless link layer Header.

Protocol	Payload[bytes]	Overhead[%]
VNC RFB	8	83.33
Microsoft RDP	6	86.96
Citrix ICA	6	86.96

Ensuring Crisp Interaction Response

Interaction latency, i.e. the delay a user experiences between generating some user input and having the result presented on his display, is key challenge of mobile cloud computing. Whereas bandwidth limitations are likely to disappear with technological advancements, interaction latency is an intrinsic key challenge of mobile cloud computing because even the most trivial user operations need to be communicated to the server. Tolia et al. point out that these trivial interactions such as moving the pointer to draw a line or to select some text, are far more challenging than loosely coupled tasks, such as web browsing. The major difference is that users expect an immediate visual result of trivial operations, whereas they anticipate processing and download delays when clicking on a link. Accustomed to the responsive interfaces of desktop applications, users will expect the same interactivity in a mobile cloud computing setting.

Remote display protocol data needs to traverse numerous links, both wireless and wired, and numerous network elements, each introducing additional propagation and transmission delays on the end-to-end path. Loss correcting retransmissions on the wireless link, router queuing, suboptimal routing schemes and firewall processing entail important propagation delays. Bandwidth limitations on the wireless link induce additional transmission delays, especially for immersive applications such as virtual environments that transfer highly detailed graphics to the client. Sometimes several client-server interactions are required before a display update can be shown on the screen, e.g. when the server waits for the acknowledgement of the client before sending the remainder of the data.

Due to the limitations in mobile bandwidth and mobile device memory resources, it is in most cases unfeasible to stream in advance all possible next display updates. Furthermore, the gains of this pre computing technique highly depend on the prediction accuracy.

A better strategy might be to buffer only a number of key display updates, for which the server only needs to provide a differential update. Boukerche et al. have evaluated a number of cache management strategies and are able to reduce the amount of requests during a 300-step movement in a 3D virtual environment from 300 to 145. Of course, in this case, the server response is still required to update the display. For more static applications, e.g. office applications, the potential next updates can be more accurately predicted as, for example, the layout of a menu will almost never change.

Consequently, the number of corrective server updates will be more limited. One typical example would be the list of recently opened files in the File menu of a text editor. Scene description languages such as MPEG-4 BiFS are particularly suited to support this client side handling of user input. The client not only receives graphic updates, but is also informed on the structure of the displayed scene and its composing objects, as well as on how the user can manipulate these objects.

IV. Conclusion

Thus a system to remotely access a computer desktop using only a cellular phone, despite the distance constraints, physical and bandwidth limitations of cellular phones is successfully proposed. The main concept of this project is to Access the information present in the PC from anywhere. This can save the valuable time. We can also access the applications in our PC from the mobile device from anywhere. Distance constraint as in the case of Blue tooth connectivity from a mobile to computer is eliminated. 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. 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.

Future Enhancements

In this paper we discussed about increasing the battery lifetime and limited bandwidth availability and interaction latency. In Future the battery life time will be increase further in devices. And also we can improve the bandwidth availability to some extent with traffic enhancements and also the interaction latency by better throughput and the turnaround time. The video audio can also viewed in an efficient way

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 TECHNICAL 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, "CrossLayer 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 2010, pp. 992–7.

[6] M. Mitrea, P. Simoens, B. Joveski, J. Marshall, A. Taguengayte, F. Preteux, 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 2009, p. 74440F (8 pp.).

[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.